

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS**  
**UNIDADE ACADÊMICA DE GRADUAÇÃO**  
**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**RAUL ANTONIO CORTIANA NETO**

**UMA FERRAMENTA DE SUPORTE À MODELAGEM COLABORATIVA DE  
SOFTWARE**

**São Leopoldo**  
**2015**

## UMA FERRAMENTA DE SUPORTE À MODELAGEM COLABORATIVA DE SOFTWARE

Raul Antonio Cortiana Neto<sup>1</sup>

Prof. Dr. Kleinner Silva Farias de Oliveira<sup>2</sup>

**Resumo:** Neste artigo apresenta-se uma análise dos trabalhos relacionados e apresentados como um estudo aprofundado, percebe-se que o meio corporativo carece de ferramentas que ofereçam suporte a modelagem colaborativa de software. Para minimizar o risco de perda de informação, conflitos de versão e a ocorrência de ruídos na comunicação, o gerenciamento das atividades do projeto necessita de um maior esforço, afim de garantir que a entrega do produto final ofereça a qualidade esperada dentro do tempo estimado e do custo previsto. Após essa análise, desenvolve-se uma ferramenta que suporta a modelagem colaborativa de software. A ferramenta avaliada deu-se através de um questionário na escala Likert por usuários da área de tecnologia da informação, na qual os resultados positivos em relação ao suporte a colaboração, mostram as melhoras na produtividade e eficiência das atividades. Sugere-se, para trabalhos futuros, a adição de novas funcionalidades na ferramenta e a utilização da mesma em um meio corporativo para realização de um estudo de caso.

**Palavras-chave:** JointJS. TogetherJS. Artefatos de Software. Colaboração

### 1 INTRODUÇÃO

Através do processo de globalização e de constantes inovações tecnológicas é notório que a demanda no setor de desenvolvimento de software tem crescido frequentemente. Com esse aumento excessivo de demandas na área de software, alternativas são implantadas para obter ganhos, tanto de tempo quanto de qualidade do processo de desenvolvimento (Nunes, 2005).

A utilização de sistemas colaborativos em processos de desenvolvimento de software surgem como uma excelente alternativa para a empresa, pois quando utilizados agregam muitos benefícios para a mesma. Pode-se dizer que esses benefícios são: Melhorias na comunicação entre os participantes envolvidos nos projetos, independente de sua localização ou fuso horário, promove a integração entre os membros da equipe. O trabalho em grupo, no meio colaborativo, promove maior facilidade na identificação e na rastreabilidade dos problemas e também

---

<sup>1</sup> Formando do curso de Análise e Desenvolvimento de Sistemas da Universidade do Vale do Rio dos Sinos – UNISINOS. raulantonio657@gmail.com

<sup>2</sup> Professor na Universidade do Vale do Rio dos Sinos – UNISINOS. kleinnerfarias@unisinos.br

proporciona maior eficiência na tomada de decisões e na disseminação de novas ideias (Mistrík et al., 2010).

Atualmente, é notório que as empresas estão buscando a utilização de softwares colaborativos constantemente, pois esses proporcionam uma interface em que os envolvidos no ciclo de vida do software possam acompanhar em tempo real e também de forma simultânea as alterações que estão ocorrendo no projeto.

A partir desses benefícios do software colaborativo, as atividades também podem ser sincronizadas entre os envolvidos no projeto o que permite um ganho de tempo no ciclo de desenvolvimento no produto e consequentemente ocasiona a redução de custo no projeto. A metodologia de trabalho colaborativo proporciona a integração entre os membros da empresa, mas também é necessário um grande esforço para manter o gerenciamento e a coordenação da equipe (Nunes, 2005).

Devido a carência de ferramentas de modelagem de artefatos de software colaborativa no mercado, é necessário que os integrantes do projeto adotem medidas adequadas visando garantir que o artefato que sofreu alterações seja comunicado. É necessário, também garantir que não aconteçam conflitos de versão e a perda de informação entre os artefatos de software. O esforço de gerenciamento em cima das atividades dos colaboradores acaba sendo maior, para que, o projeto seja executado dentro do prazo e das normas de qualidade esperadas pelo cliente em cima do produto final. (Mistrík et al., 2010).

Após uma análise dos trabalhos relacionados (apresentada na Seção 5), percebe-se que as ferramentas de modelagem carecem de algumas funcionalidades essenciais para dar suporte a colaboração para o meio corporativo, uma delas é a alteração simultânea de um artefato de software por mais de um usuário em tempo real. Este tipo de funcionalidade é hoje suportada por ferramentas de colaboração de documentos como, por exemplo, o Google docs. Se este suporte à propagação em tempo real das alterações realizadas em paralelo por diferentes usuários não é fornecido, então os modelos acabam ficando desatualizados e dessincronizados. Ainda pior, tais alterações realizadas em paralelo usualmente são contraditórias.

Analisando a literatura verifica-se que não há informações suficientes a respeito da metodologia de desenvolvimento de um sistema que suporte a modelagem colaborativa de software, dados os seguintes motivos: (1) carência de estudos e pesquisas sobre essa área; (2) atualmente não existe técnica de modelagem de artefatos de software colaborativo; (3) as ferramentas de modelagem

existentes não oferecem um bom suporte à modelagem colaborativa; e (4) na literatura, sobre esse tema, não há relatos de estudos experimentais comprovados.

Portanto, os objetivos deste trabalho são: (1) fazer uma análise comparativa das ferramentas disponíveis na literatura; (2) aprofundar os conhecimentos sobre técnicas de modelagem de artefatos de software colaborativo; (3) desenvolver uma ferramenta colaborativa para modelagem de artefatos de software; e, por fim, (4) avaliar a ferramenta proposta. Dessa forma, as contribuições deste trabalho são: (1) uma análise comparativa de ferramentas de modelagem colaborativa; (2) uma ferramenta de modelagem colaborativa de diagramas de classes da UML; e (3) uma avaliação inicial sobre os benefícios da ferramenta proposta.

Este artigo é organizado da seguinte forma. O capítulo 1 é composto pela introdução. O capítulo 2 apresenta a fundamentação teórica, discutindo os principais conceitos utilizados ao longo do trabalho. O capítulo 3 descreve o projeto da ferramenta proposta. O capítulo 4 introduz a avaliação da ferramenta proposta. O capítulo 5 apresenta uma análise comparativa de ferramentas de modelagem de software. Por fim, o capítulo 6 descreve as considerações finais e os trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 UML – Unified Modeling language

A Unified Modeling Language (UML), segundo (Booch et al., 2005), representa várias das melhores práticas de modelagem de artefatos de software, estas, ao longo do tempo, provaram que até mesmo em sistemas grandes e complexos tem obtido sucesso. A UML é uma linguagem que tem como principais objetivos, a especificação, a visualização, a construção e a documentação de artefatos de software.

A UML também pode ser utilizada em outros tipos de sistemas, como modelagem de negócios. Muitas empresas utilizam a UML para criação de seus artefatos de software, sendo que os mesmos tipos de diagramas podem ser utilizados em contextos de processos diferentes, ou seja, definir um processo padrão na organização não é um objetivo da UML, mas ela pode ser adaptada a esses processos (Booch et al., 2005).

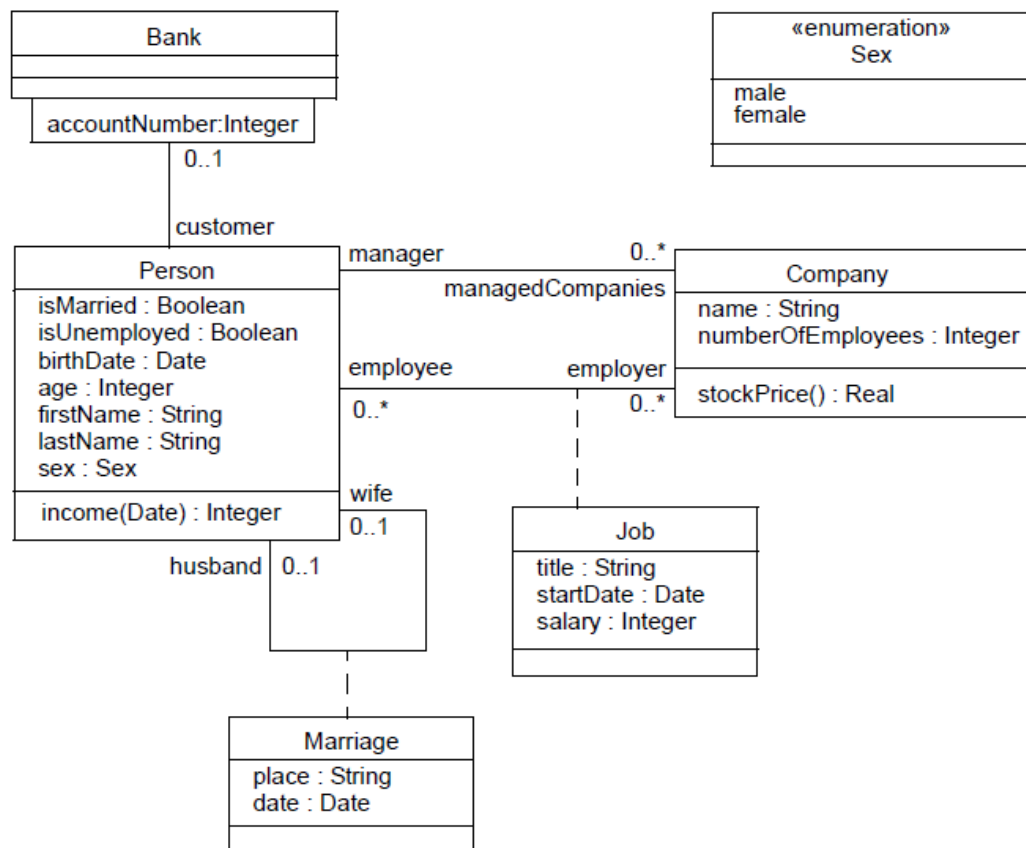
A primeira versão da Ferramenta de Modelagem de Artefatos de Software Colaborativo, da qual será detalhada no 3º capítulo deste documento, será capaz de realizar a colaboração de diagramas de classes da UML entre usuários. O Diagrama de Classes é responsável por mostrar os elementos ligados por suas diferentes estruturas estáticas. Além de conter classes com atributos e métodos o Diagrama de Classes, também pode conter interfaces, pacotes e até mesmo instâncias como objetos e links (Booch et al., 1998).

**Classe:** uma classe representa um conjunto de objetos com estruturas semelhantes juntamente com os seu comportamento e relacionamento. A UML oferece notações para caracterizar uma classe, ou seja, auxilia na definição e especificação da classe (Booch et al., 1998).

**Atributo:** os atributos representam as propriedades dos objetos das classes. Cada objeto pode ter valores específicos para cada atributo. Uma classe pode conter quantos atributos forem necessários (Booch et al., 1998).

**Método:** também conhecido como operação, os Métodos são serviços que podem ser chamados por uma instância da classe para serem utilizados no programa. Os métodos são compostos por nome e lista de parâmetros. Não há limites de Métodos a serem criados dentro de uma Classe (Booch et al., 1998).

Figura 1 – Exemplo de Diagrama de Classes



Fonte: ISO/IEC 19501(2005).

## 2.2 Modelagem Colaborativa

A colaboração, segundo (Mistrík et al., 2010), tem sido um conceito fundamental desde o início da área de engenharia de software. Nas fases iniciais, a engenharia de software não possuía ferramentas, técnicas e também processos adequados para suportar a colaboração, o que existia era muito limitado e não cobria as necessidades existentes da época. Os primeiros esforços para que houvesse colaboração em uma empresa, foram estabelecidas as seguintes características: Centralização das atividades de software, ferramentas iniciais de controle de versão, uma rígida especialização era cobrada do time e processos baseados na metodologia waterfall.

Com a introdução de ferramentas na engenharia de software e ambientes de desenvolvimento integrados, foi possível que a colaboração se tornasse um conceito mais acessível para as empresas. Atualmente, é notável que a colaboração está

sofrendo um crescimento constante, pois, principalmente em empresas multinacionais, a terceirização de serviços, projetos de software opensource, processos globais de engenharia de software, ferramentas de suporte não centralizadas e equipes distribuídas tornam-se cada vez mais comuns nas atividades diárias das empresas de desenvolvimento de software. Com isso a necessidade de utilizar ambientes e ferramentas colaborativas se faz de caráter essencial e obrigatório, principalmente em empresas globais. (Mistrík et al., 2010).

Segundo (Mistrík et al., 2010) os desafios principais que são encontrados pela engenharia de software colaborativa são as necessidades de compartilhamento de artefatos ou objetos de software e também a organização dos processos de desenvolvimento, em outras palavras, a comunicação e a coordenação das atividades e tarefas. É importante também frisar que, o processo de desenvolvimento de software juntamente com a área de gestão de projetos devem adequar-se a uma metodologia de trabalho específica, na qual suportem atividades colaborativas para cada equipe.

Como já citado anteriormente, a engenharia de software contempla a criação de vários artefatos de software, como: O produto Final, códigos e entre outros, mas também pode contemplar especificações de requisitos, modelos, planos de teste e especificação da arquitetura do sistema. Nesse contexto, podemos dizer que quando criamos cada um desses artefatos estamos realizando uma atividade colaborativa, pois vários profissionais foram engajados nessas atividades.

Uma ferramenta desenvolvida para suportar a colaboração, segundo (Mistrík et al., 2010), pode se encaixar em quatro categorias, das quais são:

- Baseada em Modelos;
- Suporte de Processos;
- Colaboração em Infra Estrutura;
- Colaboração em comunicação;

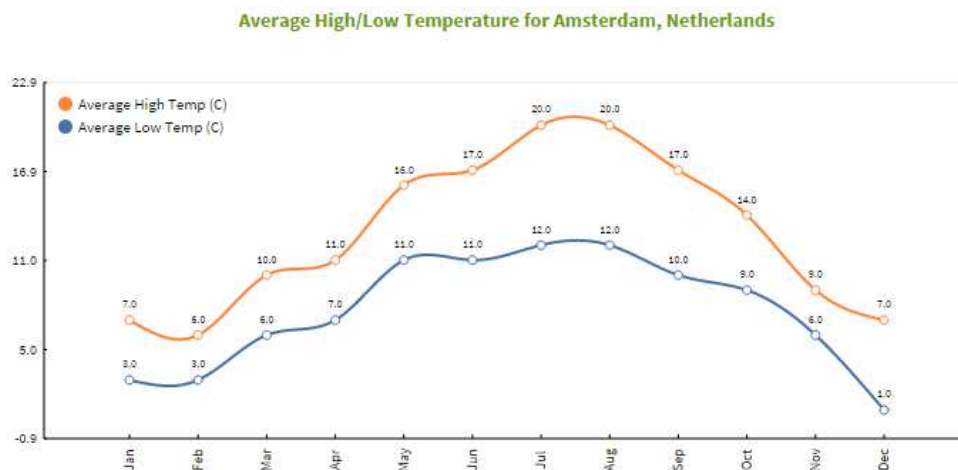
**Categoria Baseada em Modelos:** Nessa categoria, as ferramentas permitem que os engenheiros de software, em um contexto específico do software atuem de forma colaborativa, tais como em um diagrama UML por exemplo. Caracterizei essa categoria, pois nessa monografia uma ferramenta colaborativa de modelagem de artefatos de software será modelada a partir dos capítulos seguintes, onde essa se encaixa perfeitamente nessa categoria baseada em modelos (Mistrík et al., 2010).

## 2.3 JointJS

O JointJS é uma API moderna e extremamente eficaz na criação de diferentes tipos de diagramas e gráficos. A partir da união do JavaScript e do HTML5, o JointJS possibilita que os diagramas sejam criados com estruturas estáticas ou até mesmo totalmente interativas. Ao se tratar de uma moderna API, além de oferecer suporte aos navegadores Google Chrome, Safari, Internet Explorer 9 ou mais atual e Opera 15 ou mais atual, ela também oferece suporte para os navegadores de dispositivos móveis como o Google Chrome e também o Safari (Client IO, 2009).

As imagens abaixo mostram exemplos de diagramas e gráficos gerados a partir da API do JointJS.

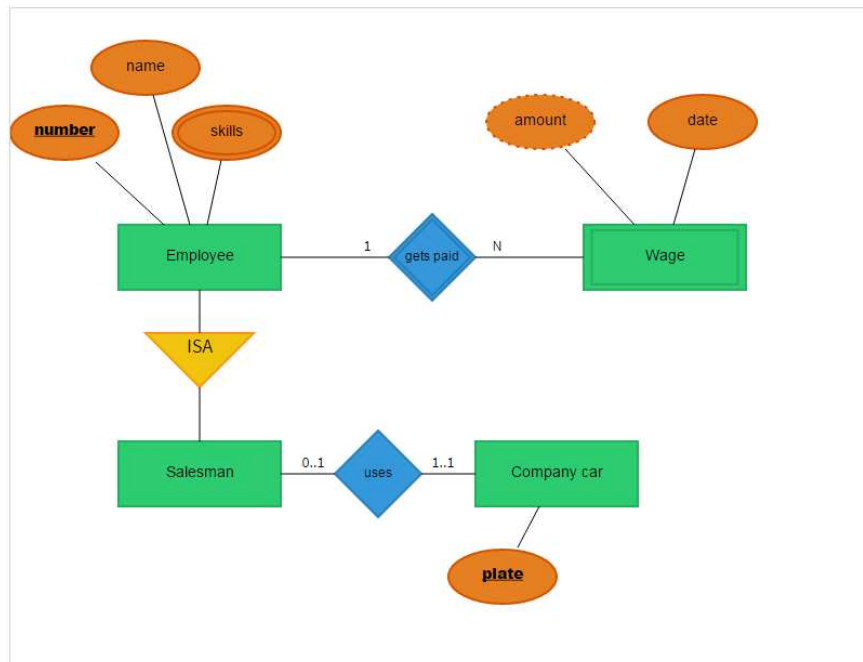
Figura 2 – Gráfico gerado a partir da API JointJS.



Fonte: <http://www.jointjs.com/demos/charts>.

Figura 3 – Diagrama gerado a partir da API JointJS.





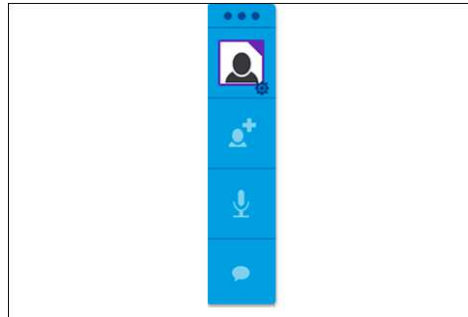
Fonte: <http://www.jointjs.com/demos/erd>.

A criação do diagrama de classes, por parte da ferramenta de modelagem de artefatos de software colaborativa, será realizada através da API do JointJS.

## 2.4 TogetherJS

A biblioteca TogetherJS foi desenvolvida com base na linguagem JavaScript pela empresa Mozilla. Esta API é gratuita e **Open Source**, esta proporciona funcionalidades e ferramentas para que seja possível que uma página web se torne eficiente em termos de colaboração. Ao adicionar essa API em um site, os usuários podem ajudar uns aos outros, de maneira colaborativa, em tempo real. O TogetherJS possui uma interface “amigável”, ou seja, sua utilização se torna simples, fácil e rápida aos olhos do usuário (Mozilla, 2013).

Figura 4 – Menu da Biblioteca TogetherJS

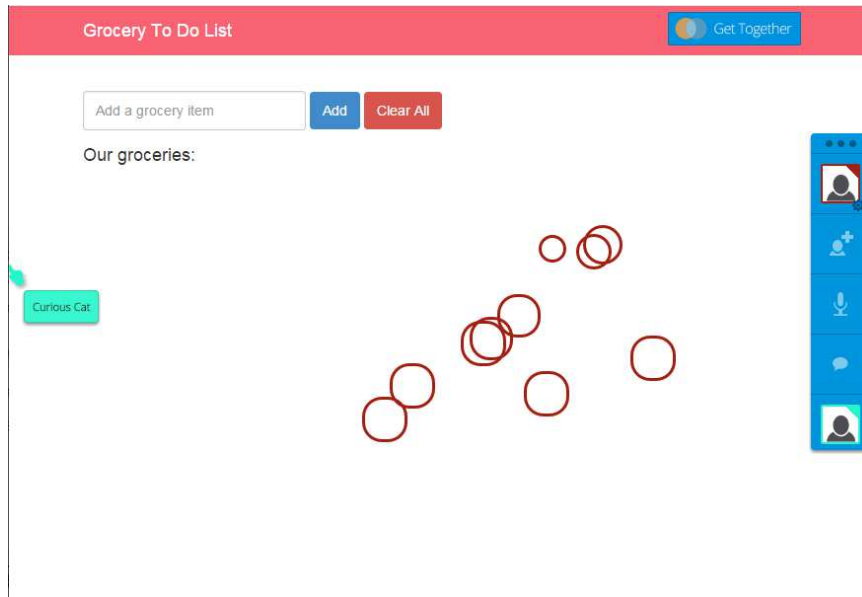


Fonte: Elaborado pelo Autor.

O TogetherJS possibilita aos usuários muitas funcionalidades em termos de colaboração, estas citadas abaixo:

- **Audio Chat:** através do uso da tecnologia Web RTC, TogetherJS possibilita a comunicação via áudio entre os usuários.
- **User Focus:** os usuários são capazes de enxergar em sua própria tela, onde o cursor dos outros usuários estão e também onde os mesmo efetuaram a função de click.
- **User Presence:** através dessa funcionalidade os usuários conseguem visualizar outros usuários em tempo real.
- **Text Chat:** Além da comunicação via Áudio também é possível a comunicação via chat.
- **Co-Browsing:** Um usuário pode acompanhar o outro usuário na navegação de uma página de um mesmo domínio.
- **Real-Time Content Sync:** Os usuários podem o ver conteúdo em um site ou até mesmo alterar esse conteúdo juntos de um modo dinâmico.

Figura 5 – Exemplo TogetherJS ativo.



Fonte: <https://togetherjs.com/>

A ferramenta de modelagem colaborativa descrita neste artigo utilizará a API TogetherJS para realizar o processo de colaboração.

### 3 FERRAMENTA PROPOSTA

#### 3.1 Requisitos do Sistema

Esta seção será composta pela definição dos requisitos técnicos e funcionais da ferramenta de modelagem de artefatos de software colaborativo.

Tabela 1. Requisitos da Ferramenta

| Sequência do Requisito | Nome do Requisito              | Descrição do Requisito   |
|------------------------|--------------------------------|--|
| REQ-001                | Alterações em Tempo Real.      | Os usuários devem ser capazes de realizar alterações no mesmo diagrama e visualizar as alterações em tempo real. |
| REQ-002                | Cursors dos usuários com foco. | Os usuários devem ser capazes de visualizar os cursores e os cliques dos   |

|         |   |  |
|---------|---|--|
|         |   | usuários da mesma seção.   |
| REQ-003 | Comunicação via chat.                                     | Deverá estar contido no sistema um chat para a comunicação dos usuários.   |
| REQ-004 | Comunicação via áudio.                                    | A ferramenta deverá ser composta por um chat de comunicação via áudio.   |
| REQ-005 | Visualização dos usuários                                 | A ferramenta deverá ser composta por uma funcionalidade onde será possível ver os usuários que estiverem “logados” na mesma seção. |
| REQ-006 | Navegação em conjunto.                                    | Se um usuário optar por mudar a página os outros usuários serão direcionados para a nova página, desde que seja no mesmo domínio.  |
| REQ-007 | Diagrama de Classes                                       | O sistema irá permitir a edição de um diagrama de classes em tempo real por todos os usuários “logados” na mesma seção.            |
| REQ-008 | Iniciar colaboração                                       | A ferramenta deve ser composta por um botão, onde este deverá iniciar o menu de colaboração.                                       |
| REQ-009 | Performance – Tempo ao adicionar um usuário para a seção. | Ao convidar um usuário para a seção, o sistema não poderá demorar mais de 5 segundos para adicionar o mesmo.                       |
| REQ-010 | Performance – Tempo para iniciar o menu de colaboração.   | Ao iniciar o menu de colaboração, o sistema não poderá demorar mais de 5 segundos para iniciar o mesmo.                            |
| REQ-011 | Performance – Tempo para carregar os recursos da página   | Ao iniciar a página do sistema o diagrama de classes deverá aparecer em menos de 7 segundos.                                       |

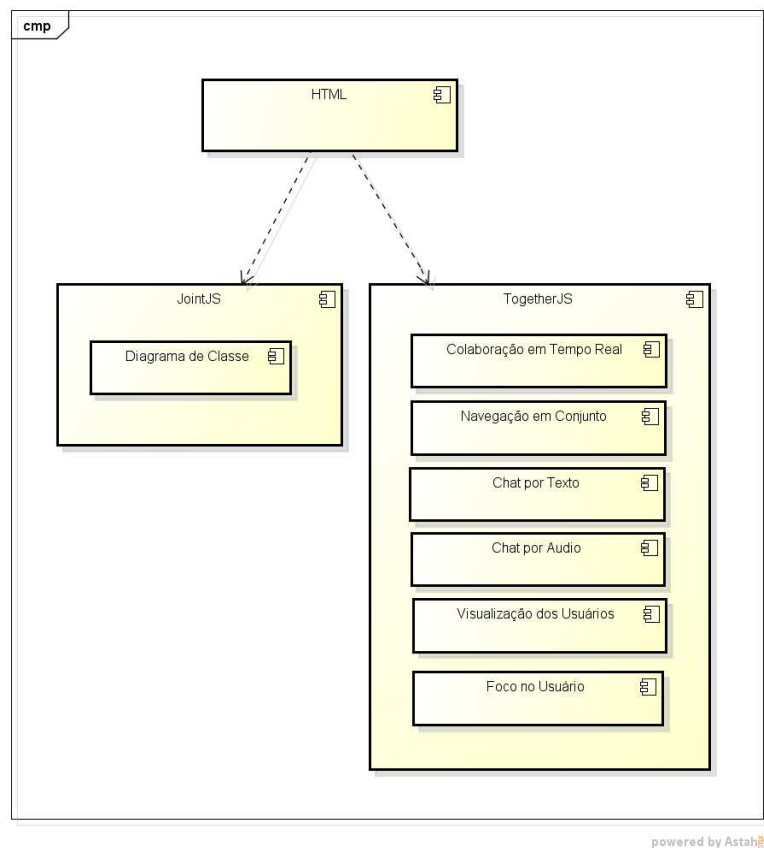
|  |                     |  |
|--|---------------------|--|
|  | ao iniciar a mesma. |  |
|--|---------------------|--|

Fonte: Elaborado pelo Autor.

### 3.2 Arquitetura da Ferramenta

Esta seção será composta pelo diagrama de componentes da ferramenta e a descrição de cada componente.

Figura 6 – Diagrama de Componentes da Ferramenta



Fonte: Elaborado pelo Autor.

A partir do diagrama acima, sigo com a descrição dos componentes do mesmo.

**HTML:** este componente é responsável por gerenciar a página inicial da ferramenta, onde está trata-se de uma ferramenta web. A parte do HTML é responsável por chamar a bibliotecas JavaScript JointJS e TogetherJS.

**TogetherJS:** componente responsável por atribuir colaboração na página web da ferramenta e também gerenciar as principais funcionalidades colaborativas oferecidas pela API.

**Colaboração em Tempo Real:** componente responsável por gerenciar as funcionalidades de colaboração em tempo real da ferramenta. Com este componente é possível que os usuários consigam utilizar a ferramenta ao mesmo tempo, sendo essas alterações visíveis e editáveis a todos os usuários.

**Navegação em Conjunto:** através desse componente, é possível que a navegação, dentro do mesmo domínio, seja seguida pelos outros usuários, ou seja, assim que um usuário efetue uma ação de navegação na ferramenta, o mesmo será seguido pelos outros usuários automaticamente.

**Chat por texto:** componente responsável por gerenciar a funcionalidade de *chat* através do modo texto com a identificação de cada usuário participante.

**Chat por Áudio:** componente responsável por gerenciar a funcionalidade de *chat* através de áudio. Essa funcionalidade se baseia na tecnologia Web RTC para disponibilizar *chat* de Áudio entre os usuários.

**Visualização dos Usuários:** este componente permite a possibilidade de identificação de cada usuário a partir de um perfil, onde o mesmo pode fazer o upload de uma imagem e alterar outras informações para definir seu perfil, como nome e cor.

**Foco no Usuário:** componente responsável por gerenciar as ações de cada usuário, mostrando onde o cursor de cada usuário está posicionado e também cada ação de clique dos usuários.

**JointJS:** componente responsável por atribuir a API de gerenciamento de diagramas e gráficos. Para essa ferramenta será abordado o Diagrama de Classe.

**Diagrama de Classe:** componente que permite a edição do diagrama de classes da ferramenta através da API JointJS.

### 3.3 Implementação

A API TogetherJS, como já citado anteriormente, atribui colaboração de uma forma muito eficiente para as páginas web. Quando a página web é criada por HTML apenas, é suficiente que a API seja apenas “chamada” dentro do código para que todos os seus recursos funcionem perfeitamente. Quando é apresentado um conteúdo dinâmico através de JavaScript, o TogetherJS necessita que uma função seja implementada para que a alteração do conteúdo seja visualizada por outros usuários, vide figura 7 abaixo.

Figura 7 – Exemplo de utilização do TogetherJS.hub.on.

```
graph.on('change:position', function(cell) {
    var _seletor = '[model-id=' + cell.attributes.id + ']',
        _posicao = 'translate(' + cell.attributes.position.x + ',' + cell.attributes.position.y + ')';

    if (TogetherJS.running) {
        TogetherJS.send({type: "changePosition", seletor: _seletor, posicao: _posicao });
    }
});

TogetherJS.hub.on("changePosition", function (msg) {
    $(msg.seletor).attr('transform', msg.posicao);
});
```

Fonte: Elaborado pelo Autor.

A ferramenta que está sendo detalhada neste artigo se encaixa perfeitamente nesse contexto, pois o diagrama de classe é criado com base na API JointJS. O **TogetherJS.hub.on** teve que ser implementado para que fosse possível que os usuários enxergassem as alterações em tempo real no diagrama de classe.

Em relação à API JointJS, como já descrito em outra seção anterior deste artigo, permite a criação de diagramas e gráficos. Para a ferramenta descrita nesse artigo será possível à edição de um diagrama de classes através dessa API. A construção do diagrama de classes da ferramenta foi efetuada pelo seguinte código mostrado na Figura 8 abaixo.

Figura 8 – Código responsável por criar o diagrama de classes.

```
var uml = joint.shapes.uml;

var classes = {

  mammal: new uml.Interface({
    id: 'interface',
    position: { x:300 , y: 50 },
    size: { width: 240, height: 100 },
    name: 'Mammal',
    attributes: ['dob: Date'],
    methods: ['+ setDateOfBirth(dob: Date): Void', '+ getAgeAsDays(): Numeric']
  )),

  person: new uml.Abstract({
    id: 'abstract',
    position: { x:300 , y: 300 },
    size: { width: 240, height: 100 },
    name: 'Person',
    attributes: ['firstName: String', 'lastName: String'],
    methods: ['+ setName(first: String, last: String): Void', '+ getName(): String']
  )),

  bloodgroup: new uml.Class({
    id: 'bloodgroup',
    position: { x:20 , y: 190 },
    size: { width: 220, height: 100 },
    name: 'BloodGroup',
    attributes: ['bloodGroup: String'],
    methods: ['+ isCompatible(bG: String): Boolean']
  )),

  address: new uml.Class({
    id: 'address',
    position: { x:630 , y: 190 },
    size: { width: 160, height: 100 },
    name: 'Address',
    attributes: ['houseNumber: Integer', 'streetName: String', 'town: String', 'postcode: String'],
    methods: []
  )),

  man: new uml.Class({
    id: 'man',
    position: { x:200 , y: 500 },
    size: { width: 180, height: 50 },
    name: 'Man'
  )),

  woman: new uml.Class({
    id: 'woman',
    position: { x:450 , y: 500 },
    size: { width: 180, height: 50 },
    name: 'Woman',
    methods: ['+ giveABrith(): Person []']
  })
}
```

Fonte: <http://jointjs.com/demos/umlcd>.

Maiores detalhes sobre a ferramenta serão abordados na próxima seção desse artigo.

### 3.4 Descrição da Ferramenta

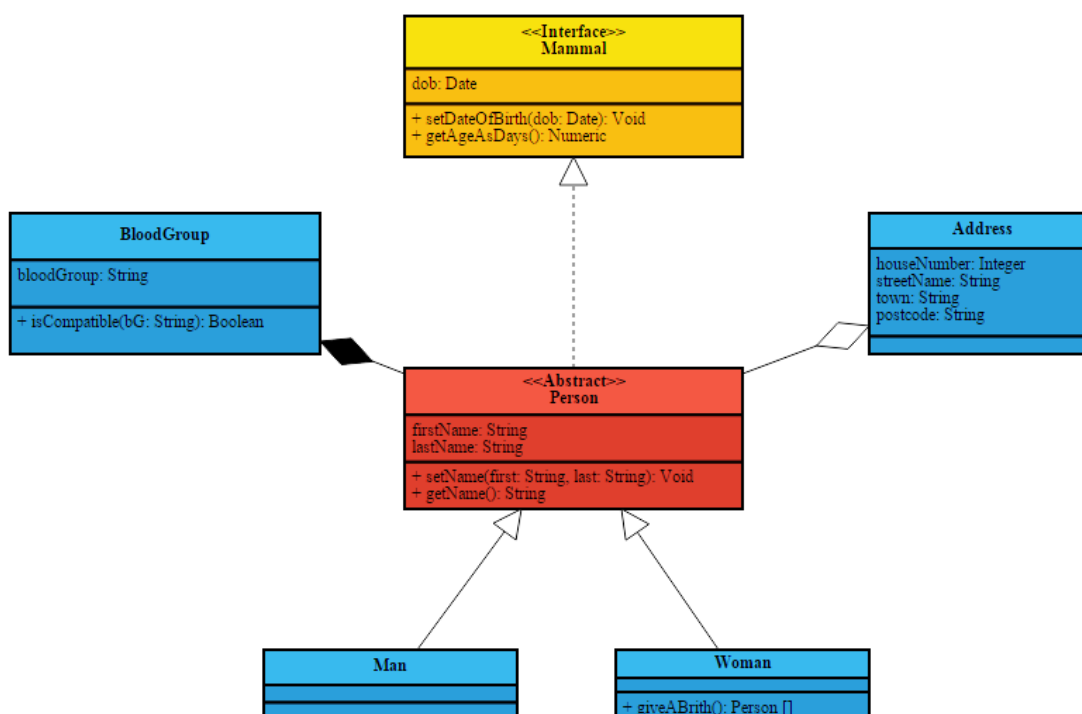
A ferramenta web de modelagem de artefatos de software colaborativo tem como principal objetivo a edição do mesmo diagrama de classe em tempo real por



um ou mais usuários de modo simultâneo. A ferramenta também dispõe de outras funcionalidades colaborativas, como comunicação por meio de chat textual e de um canal de comunicação via áudio, localização dos usuários dentro da página através da visualização do cursor e da ação de clique de cada usuário, identificação de cada usuário da mesma seção e também a navegação em conjunto dentro do mesmo domínio. Cada uma dessas funcionalidades será detalhada dentro deste capítulo.

A ferramenta está disponível em: <http://www.projetofinal.esy.es>.

Figura 9 – Diagrama de classes disponível na ferramenta.



Fonte: Elaborado pelo Autor.

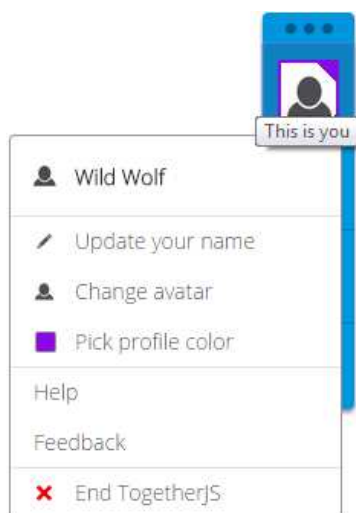
O diagrama na figura 9, mostrado acima, é gerado inicialmente na ferramenta e pode ser alterado ao mesmo tempo pelos usuários pertencentes à mesma seção.

O menu colaborativo da ferramenta possui as seguintes funções:

**Identificação do Usuário:** essa funcionalidade permite ao usuário alterar as características do seu perfil, como: Trocar o nome de seu perfil, atualizar a foto de identificação do usuário e escolher a cor mais adequada ao seu perfil. As

informações referentes ao perfil do usuário serão visualizadas por todos os usuários da mesma seção. Segue abaixo imagem de demonstração da funcionalidade.

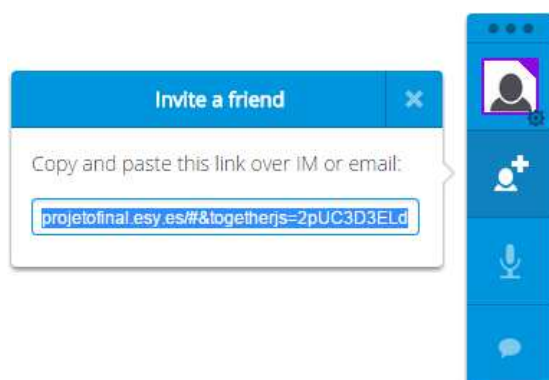
Figura 10 – Demonstração da funcionalidade Identificação do Usuário.



Fonte: Elaborado pelo Autor.

**Convidar um Usuário:** essa funcionalidade, quando ativada, gera um link para convidar um outro usuário para sua seção. Quando o outro usuário recebe esse link, basta apenas que este utilize o link no navegador para ser adicionado a mesma seção do outro usuário. A imagem abaixo mostra o link gerado pelo menu colaborativo.

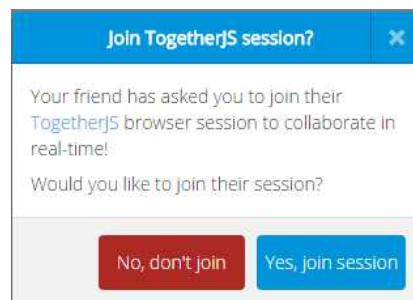
Figura 11 – Link gerado para convidar outro usuário a seção.



Fonte: Elaborado pelo Autor.

O usuário, do qual recebeu o link, ao executar esse link no navegador o mesmo receberá uma notificação perguntando se deseja entrar na seção, vide imagem abaixo.

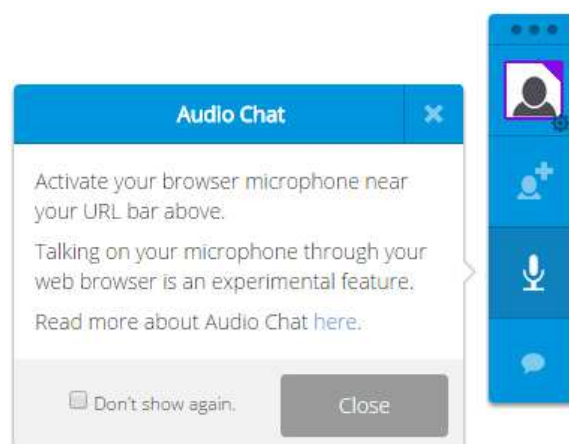
Figura 12 – Solicitação de convite ao usuário.



Fonte: Elaborado pelo Autor.

**Áudio Chat:** Essa modalidade de chat permite que os usuário conectados possam se comunicar entre si através de Áudio. Essa funcionalidade também pertence ao menu colaborativo da API TogetherJS juntamente com as outras duas funcionalidades descritas acima.

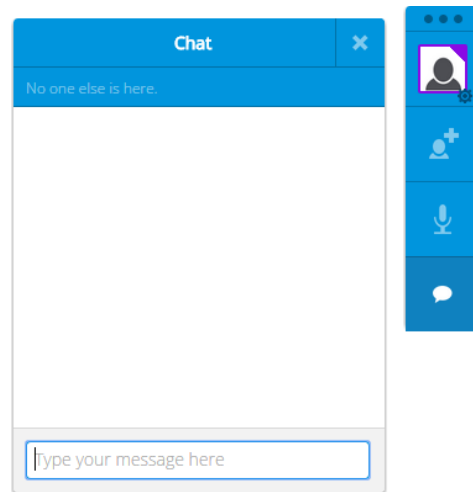
Figura 13 – Funcionalidade Audio Chat.



Fonte: Elaborado pelo Autor.

**Text Chat:** Pertencente também ao menu colaborativo, essa funcionalidade possibilita aos usuários outra opção de chat, este por um método convencional, o modo texto.

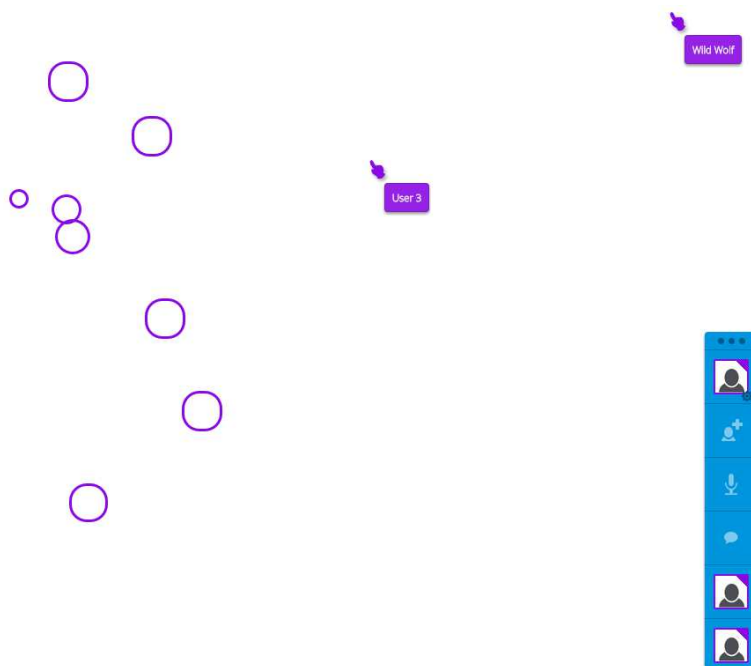
Figura 14 – Funcionalidade Text Chat.



Fonte: Elaborado pelo Autor.

A próxima imagem tem como principal objetivo mostrar a utilização da ferramenta com mais de um usuário mostrando a funcionalidade de **Foco no Usuário**.

Figura 15 – Funcionalidade de Foco no Usuário.



Fonte: Elaborado pelo Autor.

A imagem acima mostra a ferramenta sendo utilizada por três usuários ao mesmo tempo e também nos mostra que é possível localizar claramente os cursores de cada usuário. Os pequenos círculos gerados na tela mostram a ação de clique de um usuário. Essas duas características apresentadas fazem parte da funcionalidade Foco no Usuário.

#### 4 TRABALHOS RELACIONADOS

Este capítulo apresenta um quadro comparativo sobre os trabalhos estudados que focam em dar algum suporte à colaboração, sendo a maioria deles ferramentas utilizadas na indústria. Esta análise comparativa é apresentada a seguir.

- 1) ***A Web-Based, Collaborative, Computer-Aided Sequential Control Design Tool:*** esta ferramenta tem como principal objetivo realizar o *design* de um sistema de controle sequencial para dispositivos e circuitos elétricos. O software foi projetado para que vários usuários trabalhem de forma colaborativa através da internet, porém apenas o usuário dono da seção pode realizar alterações, os demais apenas podem visualizar as alterações. (Yen et al., 2003).

- 2) ***Collaborative Project Management Software***: Este artigo tem como objetivo descrever um protótipo de uma ferramenta colaborativa para gerenciamento de projetos, da qual se chama C-PMS (*Collaborative Project Management Software*). É descrito também neste artigo que a ferramenta C-PMS pode acrescentar aos envolvidos no projeto qualidades como: eficiência e eficácia nas atividades realizadas pelos mesmos (ROMANO et al., 2002).
- 3) ***Users experiences in collaborative writing using Collaboratus, an Internet-based collaborative work***: A ferramenta collaboratus é descrita no artigo como uma ferramenta para grupos de trabalho pela internet. É apresentado também suas funcionalidades colaborativas que suportam grupos de trabalho (Lowry et al., 2002).
- 4) ***IBM Rational Software Architect***: Ferramenta robusta de modelagem e design de artefatos de software que oferece suporte a colaboração. Desenvolvida pela IBM com elevado índice de reconhecimento no mercado (IBM, 2015).
- 5) ***Borland Together***: ferramenta de modelagem de software da qual permite a implementação, design e análise de arquiteturas de software. Essa ferramenta também oferece funcionalidades colaborativas aos usuários (Borland, 2015).
- 6) ***COMA - A Tool for a Collaborative Modeling***: protótipo de ferramenta para modelagem UML que possibilita suporte colaborativo para modelagem em grupos (Rittgen, 2008).
- 7) ***CoID SPA – A Tool For Collaborative Process Model Development***: protótipo de ferramenta que tem como objetivo uma prova de conceito que uma ferramenta baseada na web pode suportar modelagem colaborativa. (Lee et al., 2000).

Questões para avaliação dos trabalhos relacionados:

- Critério 1: a ferramenta permite alterações em tempo real e simultâneas no mesmo artefato de software?
- Critério 2: a ferramenta possibilita comunicação entre os usuários?
- Critério 3: a ferramenta possibilita a identificação de cada usuário?
- Critério 4: a ferramenta permite localizar a ação do usuário?

- Critério 5: a ferramenta mostra qual foi o ultimo usuário a atualizar o conteúdo?
- Critério 6: a ferramenta permite colaboração para usuários não conectados?
- Critério 7: a ferramenta possibilita utilização multi-plataforma?
- Critério 8: a ferramenta pode ser utilizada em dispositivos móveis?
- Critério 9: a ferramenta permite integração com outras ferramentas?
- Critério 10: a ferramenta permite a publicação de notícias?

A Tabela 2 apresenta um quadro comparativo entre ferramentas colaborativas, com base nos trabalhos relacionados e nos critérios de avaliação citados acima .

Tabela 2. Quadro comparativo entre as ferramentas colaborativas.

|             | Ferramentas |   |   |   |   |   |   |
|-------------|-------------|---|---|---|---|---|---|
|             | 1           | 2 | 3 | 4 | 5 | 6 | 7 |
| Critério 1  |             |   |   |   |   |   |   |
| Critério 2  | x           | x | x | x | x | x | x |
| Critério 3  | x           | x | x | x | x |   | x |
| Critério 4  |             |   |   |   |   |   |   |
| Critério 5  | x           | x |   | x | x | x | x |
| Critério 6  | x           | x | x | x | x | x | x |
| Critério 7  | x           |   |   |   |   |   |   |
| Critério 8  |             |   |   |   |   |   |   |
| Critério 9  |             |   | x | x | x |   | x |
| Critério 10 |             | x | x | x |   |   |   |

A partir da avaliação das ferramentas colaborativas na tabela 2 é evidente que, mesmo na era da colaboração, ainda exista carência de ferramentas que atendam o mercado satisfazendo 100% das necessidades colaborativas dos usuários.

## 5 AVALIAÇÃO DA FERRAMENTA

A avaliação da ferramenta foi realizada através de um questionário respondido por profissionais de diferentes áreas da TI. Este questionário está dividido em duas

partes. A primeira parte é composta por perguntas com o objetivo de determinar as características do perfil do profissional. A segunda parte tem como objetivo avaliar as funcionalidades da ferramenta de modelagem colaborativa. Esta parte do questionário foi elaborada com base na escala *Likert* (JAMIESON, 2004) que, por sua vez, proporciona aos usuários cinco alternativas para cada pergunta, essas alternativas são:

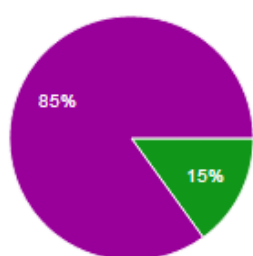
- Discordo Totalmente.
- Discordo Parcialmente.
- Não sei Opinar.
- Concordo Parcialmente.
- Concordo Totalmente.

O questionário completo se encontra no APÊNDICE A deste artigo.

A partir da avaliação da ferramenta sobre a funcionalidade do canal de comunicação (Figura 16) e segundo (Mistrík et al., 2010) conclui-se que a comunicação disponibilizada pela ferramenta para os usuários, proporciona benefícios a equipe, tais como: Melhora no entendimento da solução proposta, maior compreensão em relação ao diagrama elaborado, maior agilidade na tomada de decisões e também na disseminação de novas idéias.

Figura 16 – Questão quatro do questionário.

**4) Os canais de comunicação da ferramenta funcionam a contento?**



|                       |    |     |
|-----------------------|----|-----|
| Discordo totalmente   | 0  | 0%  |
| Discordo Parcialmente | 0  | 0%  |
| Não sei opinar        | 0  | 0%  |
| Concordo Parcialmente | 3  | 15% |
| Concordo totalmente   | 17 | 85% |

Fonte: Elaborado pelo Autor.

A questão de número cinco (Figura 17) do questionário avaliativo, nos mostra que mais de 80% dos profissionais que responderam ao questionário, concordam que a ferramenta minimiza a probabilidade de conflito entre versões de artefatos.

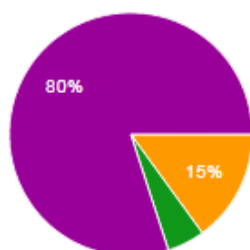


Isso deve-se ao fato de que a ferramenta possibilita alterações no mesmo diagrama por múltiplos usuários em tempo real e de maneira simultânea.

Segundo (Mistrík et al., 2010) as atividades sincronizadas entre os envolvidos no projeto permitem ganhos em relação a redução do tempo de desenvolvimento, o que ocasiona uma redução no custo do mesmo, além dos benefícios já citados no parágrafo anterior.

Figura 17 – Questão cinco do questionário.

**5) A ferramenta minimiza a probabilidade de conflitos entre as versões?**



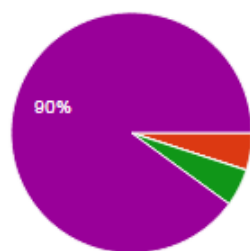
|                       |    |     |
|-----------------------|----|-----|
| Discordo totalmente   | 0  | 0%  |
| Discordo Parcialmente | 0  | 0%  |
| Não sei opinar        | 3  | 15% |
| Concordo Parcialmente | 1  | 5%  |
| Concordo totalmente   | 16 | 80% |

Fonte: Elaborado pelo Autor.

Avaliando a questão dois do questionário (Figura 18), e com base a todos os benefícios já citados anteriormente, é notório que a ferramenta colaborativa proporciona um aumento de produtividade considerável em relação as atividades por ela realizada. Segundo (Mistrík et al., 2010) as ferramentas colaborativas e em especial a ferramenta de modelagem colaborativa descrita nesse artigo, diminuem a necessidade de compartilhar artefatos de software e também proporcionam a eficiência na comunicação e na coordenação das atividades e trabalhos.

Figura 18 – Questão dois do questionário.

**2) As funcionalidades colaborativas proporcionam maior produtividade?**



|                       |    |     |
|-----------------------|----|-----|
| Discordo totalmente   | 0  | 0%  |
| Discordo Parcialmente | 1  | 5%  |
| Não sei opinar        | 0  | 0%  |
| Concordo Parcialmente | 1  | 5%  |
| Concordo totalmente   | 18 | 90% |

Fonte: Elaborado pelo Autor.

## **6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS**

Através da realização deste artigo e da implementação e avaliação da ferramenta de suporte a modelagem colaborativa de software, é notório que a colaboração proporciona muitos benefícios a empresa e conseqüentemente ao produto final, como: atividades podem ser realizadas em sincronia, ocasionando diminuição do tempo na execução das atividades e conseqüentemente a redução do custo do projeto; minimiza conflitos de versão e também a perda de informação entre artefatos de software; possibilita melhorias nos canais de comunicação dos envolvidos no projeto; facilita na identificação e na rastreabilidade de problemas; proporciona maior eficiência na tomada de decisões e na disseminação de novas ideias; e por fim promove a integração da equipe.

Atualmente, pode-se dizer que, a colaboração é uma característica chave, os benefícios citados no parágrafo acima se fazem presentes na lista de melhorias na maioria das empresas, contudo, ainda permanecem carências por parte das ferramentas colaborativas responsáveis por agregar esses benefícios nas empresas. A maioria das ferramentas ainda não satisfazem as necessidades reais de colaboração das empresas e necessita-se estudos mais aprofundados referentes a metodologias de trabalho com ferramentas colaborativas.

Sugere-se como trabalhos futuros, a ampliação das funcionalidades da ferramenta juntamente com uma nova avaliação da mesma, para que a mesma consiga satisfazer ainda mais as necessidades colaborativas existentes. Sugere-se também como outro tópico a ser trabalhado, um estudo de caso com base na utilização da ferramenta para atividades reais de trabalho.

## **A TOOL TO SUPPORTING COLLABORATIVE MODELING SOFTWARE**

This paper presents an analysis of the related work and presented as a comprehensive study, it is clear that the corporate environment needs tools that support collaborative modeling software. To minimize the risk of information loss, version conflicts and the occurrence of noise in the communication, the management of project activities requires a greater effort in order to ensure that the delivery of the end product offers the expected quality within the estimated time and of the estimated cost. After this analysis, there is developed a collaborative tool that supports the modeling software. The evaluated tool was made through a questionnaire in Likert scale by users of information technology, in which the positive results regarding support collaboration, show improvements in productivity and efficiency of the activities. It is suggested, for future work, adding new features in the tool and its use in a corporate environment to conduct a case study.

**Keywords:** JointJS. TogetherJS. Software Artifacts. Collaboration

## REFERÊNCIAS

- BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. Unified Modeling Language Specification. 2005. p 355-378.
- BOOCH, G; RUMBAUGH, J; JACOBSON, I.: The Unified Modeling Language Reference Manual, Addison-Wesley, 1a edition, 1998.
- BORLAND Together. Disponível em:  
<<http://www.borland.com/Products/Requirements-Management/Together>>. Acesso em: 02 mai. 2015.
- Client IO. JointJS JavaScript Diagraming Library. Disponível em:<  
<http://www.jointjs.com>>. Acesso em 01 mar. 2015.
- ESMIN, Ahmed Ali Abdalla. Modelando com UML – Unified Modeling Language. Disponível em: < <http://www.dcc.ufla.br/infocomp/artigos/v1.1/art09.pdf>>. Acesso em: 28 nov. 2014.
- ISSO/IEC 2005. Meta Object Facility (MOF) Specification. 2005. p 13-24.
- ISSO/IEC 2014. Information technology - Object Management Group Meta Object Facility (MOF) Core. 2014. p 6-14
- ISSO/IEC 2014. Information technology - Object Management Group XML Metadata Interchange (XMI). 2014. p 5-43.
- JAMIESON, S. Likert scales: how to (ab)use them. **Medical Education**, v. 38, n. 12, [S.l.], p. 1217-1218, 2004
- LEE, James D.; HICKEY, Ann M.; ZHANG, Dongsong; SANTANEN, Eric; ZHOU, Lina. ColD SPA: A Tool For Collaborative Process Model Development. 2000. Disponível em:  
<<https://www.computer.org/csdl/proceedings/hicss/2000/0493/01/04931004.pdf>>. Acesso em: 01 mai. 2015.
- LOWRY, Paul Benjamin; ALBRECHT, Conan C.; LEE, James D. NUNAMAKER, Jay F. Users experiences in collaborative writing using Collaboratus, an Internet-based collaborative work. 2002. Disponível em:  
<<http://www.computer.org/csdl/proceedings/hicss/2002/1435/01/14350021.pdf>>. Acesso em: 01 mai. 2015
- MISTRIK, Ivan; GRUNDY, John; HOEK, André van der; WHITEHEAD, Jim. Collaborative Software Engineering. 2010. p 35-93.
- MOTA, António; VALENTE, Hugo; NAVEGA, Ivo; PACHECO, Pedro; SILVA, Tiago; PACHECO, Pedro. UML 2.0 - Uni\_ed Modeling Language 2.0. Disponível em:  
<<http://paginas.fe.up.pt/~ei02084/artigo.pdf>>. Acesso em: 28 nov. 2014.

Mozilla.TogetherJS Collaboration Made Easy. Disponível em:<  
<https://www.togetherjs.com>>. Acesso em 01 mar. 2015.

NUNES, Vanesa Batestin. Integrando Gerência de Configuração de Software, Documentação e Gerência de Conhecimento em um Ambiente de Desenvolvimento de Software. 2005. Disponível em:  
<http://inf.ufes.br/~falbo/files/DissertacaoNunesVanessa.pdf>. Acesso em: 28 nov. 2014.

OMG, Unified Modeling Language: Superstructure, Version 2.4.1, 2011.

IBM Rational Software Architect (IBM RSA). Disponível em:  
 <<http://www.ibm.com/developerworks/downloads/r/architect/index.html>>. Acesso em: 02 mai. 2015.

RITTGEN, Peter. COMA: A Tool for a Collaborative Modeling. 2008. Disponível em:  
 < <http://ceur-ws.org/Vol-344/paper16.pdf> > Acesso em: 01 mai. 2015.

ROMANO, Nicholas C; CHEN, Fang; NUNAMAKER, Jay F. Collaborative Project Management Software. 2002. Disponível em:  
 <<http://www.computer.org/csdl/proceedings/hicss/2002/1435/01/14350020b.pdf>>. Acesso em: 01 mai. 2015.

STEINBERG, Dave; BUDINSKY, Frank; PATERNOSTRO, Marcelo; MERKS, Ed. EMF – Eclipse Modeling Framework. 2008. P 28-57

VARGAS, Thânia Clair de Souza. A história de UML e seus diagramas. Disponível em: <[https://projetos.inf.ufsc.br/arquivos\\_projetos/projeto\\_721/artigo.tcc.pdf](https://projetos.inf.ufsc.br/arquivos_projetos/projeto_721/artigo.tcc.pdf)>. Acesso em: 28 nov. 2014.

YEN, Chiaming; LI, Wu-Jeng; LIN, Jui-Cheng. A Web Based, Collaborative, Computer Aided Sequential Control Design Tool. 2003. Disponível em:  
 <<http://www.ieeecss.org/CSM/library/2003/april03/2-EyeOnEducation.pdf>>. Acesso em: 01 mai. 2015

## APÊNDICE A

### Questionário sobre a Avaliação da Ferramenta

Esse questionário, tem como seu principal objetivo, a avaliação da ferramenta de suporte a modelagem colaborativa de software. A ferramenta foi implementada como parte de um trabalho de conclusão do curso de Análise e Desenvolvimento de Sistemas da Universidade do Vale do Rio dos Sinos – Unisinos.

**Título do Trabalho:** Uma Ferramenta de Suporte à Modelagem Colaborativa de Software.

**Aluno:** Raul Antonio Cortiana Neto.

**Orientador:** Prof. Dr. Kleinner Silva Farias de Oliveira.

| Caracterização do Entrevistado                                 |
|--|
| Idade:   |
| Profissão:   |
| Cargo Atual:   |
| Empresa em que Trabalha:                                       |
| Quanto tempo está nesse cargo:                                 |
| Maior grau de escolaridade:                                    |
| Qual a sua formação acadêmica:                                 |
| Por quanto tempo você estudou (tem estudado) em Universidades: |
| Seu cargo atual se encaixa mais em qual especialidade:         |
| Quanto tempo de experiência em desenvolvimento de software:    |
| Quanto tempo de experiência em modelagem de software:          |

| Avaliação da Ferramenta   |                     |                       |                |                       |                     |
|---|---------------------|-----------------------|----------------|-----------------------|---------------------|
| Questão   | Discordo Totalmente | Discordo Parcialmente | Não sei Opinar | Concordo Parcialmente | Concordo Totalmente |
| 1. A ferramenta é intuitiva?  |                     |                       |                |                       |                     |
| 2. As funcionalidades proporcionam maior produtividade?                       |                     |                       |                |                       |                     |
| 3. O tempo de sincronização das ações dos diferentes usuários é satisfatório? |                     |                       |                |                       |                     |

|   |  |  |  |  |  |
|---|--|--|--|--|--|
| 4. Os canais de comunicação da ferramenta funcionam a contento?                                   |  |  |  |  |  |
| 5. A ferramenta minimiza a probabilidade de conflitos entre as versões?                           |  |  |  |  |  |
| 6. A ferramenta possui identificação dos usuários que estão trabalhando de forma colaborativa?    |  |  |  |  |  |
| 7. É fácil a adição de novos usuários para trabalharem de forma simultânea na seção?              |  |  |  |  |  |
| 8. A ferramenta disponibiliza as ferramentas necessárias para a modelagem do diagrama de classes? |  |  |  |  |  |
| 9. É fácil identificar as ações dos usuários?   |  |  |  |  |  |
| 10. É satisfatório o tempo que um usuário leva para se conectar a ferramenta?                     |  |  |  |  |  |