

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Lucas Oliveira Costa

**Colaboração e cooperação em Arte Digital com  
o Ambiente *Mosaicode***

São João del-Rei

2019

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI

Lucas Oliveira Costa

**Colaboração e cooperação em Arte Digital com o  
Ambiente *Mosaiccode***

Monografia apresentada como requisito da disciplina de Projeto Orientado em Computação II do Curso de Bacharelado em Ciência da Computação da UFSJ.

Orientador: Flávio Luiz Schiavoni

Universidade Federal de São João del-Rei – UFSJ

Bacharelado em Ciência da Computação

São João del-Rei

2019

Lucas Oliveira Costa

## **Colaboração e cooperação em Arte Digital com o Ambiente *Mosaiccode***

Monografia apresentada como requisito da disciplina de Projeto Orientado em Computação II do Curso de Bacharelado em Ciência da Computação da UFSJ.

Trabalho aprovado. São João del-Rei, 26 de junho de 2019:

---

**Flávio Luiz Schiavoni**  
Orientador

---

**Sofia Larissa da Costa**  
Convidado 1

---

**Rogério Tavares Constante**  
Convidado 2

São João del-Rei  
2019

*Este trabalho é dedicado à meus pais, família, amigos e todos aqueles que sempre acreditaram em mim e ajudaram suportar as dificuldades.*

# Agradecimentos

A Deus, por me iluminar e dar paciência e perseverança;

Aos meus pais, que sempre me apoiaram em todos os momentos da minha vida.

Aos meus irmãos e sobrinhos, por me ajudarem incansavelmente durante os anos que mais precisei.

Aos meus amigos, minha namorada e todos que acreditaram na minha capacidade e me levantaram quando eu estava já cansado.

A todos meus familiares, que torceram pelo meu sucesso.

Ao meu orientador pela oportunidade de crescimento oferecida durante esse trabalho.

Aos professores que fizeram parte da minha construção como pessoa.

À UFSJ pelas oportunidades.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

# Resumo

A arte digital é um crescente ramo multidisciplinar na área da computação e artes. Suas peculiaridades tornam o trabalho em equipe um desafio tanto para cientistas quanto para os artistas envolvidos em projetos em conjunto sobre o tema. Este trabalho tem como objetivo estudar formas e desenvolver ferramentas para a colaboração em arte digital com o sistema Mosaiccode, trazendo conceitos de colaboração em computadores e definindo artefatos importantes no desenvolvimento e compartilhamento de arte digital.

**Palavras-chaves:** Mosaiccode. Colaboração. Arte Digital.

# Abstract

Digital art is a growing multidisciplinary field in the area of computing and the arts. Its quirks make teamwork a challenge for both scientists and artists involved in joint projects on this subject. This work aims to study methods and develop tools for collaboration in digital art with the Mosaicode system, bringing collaboration concepts supported by computers and defining important artifacts in the development and sharing of digital art.

**Key-words:** Mosaicode. Collaboration. Digital Art.

# Lista de figuras

Figura 1 – Propriedades estáticas de um Bloco Oscilador / Diagrama exemplo para um programa que gera áudio com a posição do mouse. . . . .	12
Figura 2 – Fluxo da criação de uma aplicação no <i>Mosaiccode</i> . . . . .	13
Figura 3 – Fluxograma da metodologia do POC II . . . . .	16
Figura 4 – Aplicações do Modelo 3C. . . . .	20
Figura 5 – Crianças participam de obra de arte digital. . . . .	25
Figura 6 – GitHub Manager para Processing. . . . .	29
Figura 7 – Tela de chat do protótipo desenvolvido . . . . .	32
Figura 8 – Caso de uso: peer-to-peer . . . . .	33
Figura 9 – Caso de uso: multicast . . . . .	34
Figura 10 – Mensagens multicast e unicast . . . . .	34
Figura 11 – Protocolo de comunicação proposto e desenvolvido neste trabalho . . .	36
Figura 12 – Mensagens enviadas pelo sistema na entrada de um novo membro no grupo . . . . .	37
Figura 13 – Resposta do grupo à entrada no sistema da figura 12 . . . . .	38



# Lista de tabelas

Tabela 1 – Tabela de Ferramentas de comunicação Síncrona e Assíncrona . . . . .	21
---	----

# Lista de abreviaturas e siglas

3C	Colaboração, Coordenação e Comunicação.
API	Application Programming Interface
CDE	Collaborative Development Environment
CSCW	Computer Supported Cooperative Work
CVS	Concurrent Version System
DSL	Domain-specific Language
GPL	General-purpose Language
IDE	Integrated Development Environment
IP	Internet Protocol
P2P	Peer-To-Peer
PD	Pure Data
SNV	Apache Subersion
TTL	Time To Live
UDP	User Datagram Protocol
VPL	Visual Programming Language
XML	Extensible Markup Language

# Sumário

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Ambiente Mosaicode	11
1.2	Artefatos de Software no Mosaicode	12
1.3	Colaboração em Ambientes de Software	14
1.4	Arte digital e colaboração	14
1.5	Objetivos	15
1.6	Metodologia	16
<b>2</b>	<b>Computer Supported Cooperative Work</b>	<b>18</b>
2.1	Histórico	18
2.2	Groupware	19
2.3	O ambiente dinâmico do trabalho (Interação)	19
2.4	Ferramentas	20
2.4.1	O Modelo 3C	20
2.4.2	Repositórios e Workspace	22
2.4.3	Ferramentas integradas	23
<b>3</b>	<b>Colaboração em Arte Digital</b>	<b>24</b>
3.1	Criação em Arte Digital	26
3.1.1	GPL vs. DSL	26
3.2	Ferramentas para criação de arte digital	27
<b>4</b>	<b>Prototipação</b>	<b>31</b>
4.1	Decisões de projeto	31
4.2	Resultados e implementação	33
4.2.1	Protocolos detalhados	35
4.2.2	Aplicação	37
<b>5</b>	<b>Conclusão</b>	<b>39</b>
	<b>Referências</b>	<b>41</b>

# 1 Introdução

A arte digital é um crescente ramo das artes e tecnologia. No contexto dessa interseção estão crescendo tanto o número de artistas, quanto os profissionais de tecnologia envolvidas num mesmo projeto (AHMED S.U.; TRIFONOVA; SINDRE, 2009). Com sua popularização, temos que pensar em novas possibilidades de interação e cooperação entre todos os envolvidos quebrando as barreiras da multidisciplinaridade em questão.

Os ambientes de desenvolvimento integrados auxiliam no desenvolvimento de novos sistemas pois possuem uma gama de ferramentas para apoiar o desenvolvedor a realizar sua tarefa de modo mais prática e rápida. Para o âmbito da arte digital, temos o Mosaicode, um ambiente de programação visual com o intuito de atender as demandas dos campos das artes digitais e gerar aplicações para este campo. Pretende-se que esta ferramenta ajude os artistas a desenvolver o seu trabalho sem precisar ter o conhecimento em linguagens de programação. O Mosaicode dispõe das facilidades gráficas de diagramas para representação dos objetos (através de caixas e conexões). Ao final, é gerado um código-fonte, na linguagem desejada, que realizará as funções descritas no modelo.

Entretanto, o Mosaicode não possui ainda formas de permitir trabalhos cooperativos ou colaborativos como compartilhar seus resultados e códigos. Além disso, como é um sistema de programação visual, também não conseguimos compartilhar os diagramas que geraram os programas desenvolvidos pela ferramenta. Nosso objetivo, portanto, é estudar os artefatos que são gerados no desenvolvimento dentro do Mosaicode e criar meios para se compartilhar esses artefatos.

Para permitir a colaboração será importante explicar conceitos sobre a utilização de ferramentas de colaboração entre dois computadores e sobre a comunicação entre eles. Foi desenvolvido um protocolo de rede e modificações para controle dos artefatos gerados pelo sistema. Serão abordados temas sobre toda a história da colaboração entre pessoas utilizando softwares, como também como esse compartilhamento de informações interfere na arte digital e propostas de implementação para esse tipo de ferramenta.

## 1.1 Ambiente Mosaicode

O Mosaicode é um ambiente de programação visual formado por componentes são chamados de blocos. Os blocos são as peças mínimas do fluxo de programação da ferramenta e possuem propriedades que são estáticas e dinâmicas (GONÇALVES, 2017). As estáticas, podendo ser configuradas no ambiente, são valores que foram definidas pelo gerador do código do bloco, por exemplo, o tamanho de um determinado elemento na

tela. As dinâmicas são definidas pela interação de dois blocos, através de conexões. As conexões, por sua vez, são feitas através de portas de entrada e saída de cada bloco no próprio editor do diagrama.

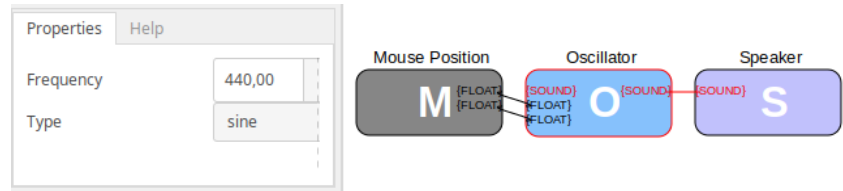


Figura 1 – Propriedades estáticas de um Bloco Oscilador / Diagrama exemplo para um programa que gera áudio com a posição do mouse.

Um diagrama é uma coleção de blocos e conexões, como o apresentado na Figura 1. Se mudarmos uma das conexões, estaremos gerando um novo diagrama, portanto um código-fonte diferente do primeiro deverá ser obtido por esse novo diagrama, com resultados possivelmente bem diferentes. Note também, que para funcionar as conexões, os valores de entrada e saída devem ser do mesmo tipo, que são definidos pelo próprio bloco gerador.

O Mosaicode é uma ferramenta extensível, ou seja, aceita novas extensões de funcionalidades. Isso implica uma maior flexibilidade (GONÇALVES, 2017) do ambiente. Por meio de extensões, podemos adicionar novos blocos, portas e padrões de códigos nas bibliotecas de blocos. Isso já pode ser feito inclusive, dentro do próprio Mosaicode plugin *Library Manager*, que também é uma extensão, e ao ser instalado, cria um novo item de menu na barra de menu do sistema.

Os componentes, sejam eles Blocos, Portas ou Padrões de Código, pode ser feitos tanto como uma classe Python quanto como um arquivo XML. Arquivos XML podem ainda estar em espaço de usuário ou instalada com o sistema. O carregamento da ferramenta se dá com o carregamento dos plugins na seguinte ordem: Classes python instaladas com o sistema, Arquivos XML instalados com o sistema e Arquivos XML no espaço de usuário. Assim, se o usuário quiser alterar e personalizar Blocos em sua instância da ferramenta, o mesmo consegue fazer de maneira a alterar a instalação da ferramenta apenas para seu usuário e sem depender de senha de acesso especial para isto. (GONÇALVES, 2017)

## 1.2 Artefatos de Software no Mosaicode

Os artefatos de software são todos os subprodutos gerados no desenvolvimento de um software, tais como, relatórios, documentação, diagramas e até mesmo o código-fonte. O gerenciamento desses artefatos podem ser providos pelo ambiente de desenvolvimento utilizado e tem extrema importância no gerenciamento do software pois os artefatos podem

ser reutilizados, modificados e atualizados e sem o controle destes artefatos o processo num ambiente colaborativo ficaria caótico (FUKS; RAPOSO; GEROSA, 2002).

Quais são os artefatos que deveremos gerenciar na ferramenta? Primeiro temos que identificar quais os artefatos são gerados pelo processo de desenvolvimento no ambiente e seu fluxo.

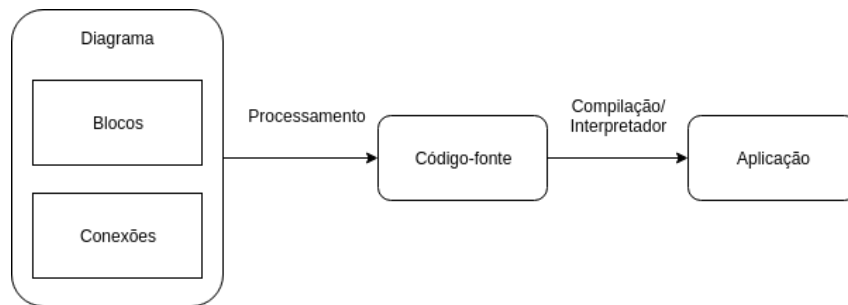


Figura 2 – Fluxo da criação de uma aplicação no *Mosaicode*.

Através do gráfico do fluxo na criação de uma aplicação, apresentado na Figura 2, podemos identificar explicitamente alguns artefatos gerados pelo ambiente:

- Diagrama: O diagrama que gera a aplicação é onde temos o resultado de um processo criativo, o programa final daquele desenvolvimento. É o principal componente que o usuário terá contato no desenvolvimento na ferramenta;
- Código-fonte: O código-fonte é um subproduto gerado à partir do processamento do diagrama. Esse, será obtido através do processamento das configurações atribuídas aos blocos e conexões, tais como os blocos de códigos que são estáticos, quanto às propriedades dinâmicas de conexões entre os blocos. Apesar de ser extremamente importante para a criação e execução da ferramenta, este código pode ser gerado diversas vezes pela ferramenta e, então, não é um artefato que o usuário cria diretamente;
- Aplicação: A partir do código-fonte gerado a ferramenta compila ou executa através de um interpretador (dependendo da linguagem destino escolhida) e gera a aplicação. Este também é um subproduto do desenvolvimento de um diagrama que não é desenvolvido diretamente pelo usuário mas gerado pelo ambiente. Esta aplicação, portanto, também será um artefato do ambiente.

Note também que o diagrama é composto por blocos e conexões. Esses blocos, como discutido no capítulo anterior, são adquiridos através das extensões e de bibliotecas de blocos. Os blocos, como são as peças mínimas da ferramenta, bem como suas coleções, por isso, implicitamente também são artefatos muito importantes do sistema.

## 1.3 Colaboração em Ambientes de Software

Os ambientes de software são ferramentas integradas que dão suporte ao desenvolvedor na criação de sua aplicação. Essas ferramentas podem ser, por exemplo, editores de textos para a codificação e facilidades de refatoração de código e depuração de erros e até mesmo telas de desenvolvimento rápido, onde o programador apenas arrasta componentes para a criação de softwares.

Quando se tem uma equipe para o desenvolvimento de aplicações, é necessário portanto, um meio de colaboração e cooperação para que os integrantes desse projeto possam desenvolver de maneira conjunta. Os ambientes portanto, precisam integrar ferramentas para que essa colaboração ocorra e de maneira eficaz e segura.

No princípio do desenvolvimento em equipe observou-se a necessidade de se criar ferramentas para que o processo de trabalho em conjunto não se tornasse um verdadeiro caos. Imagine: duas pessoas podem alterar o mesmo código, onde uma pode apagar algo importante no código feito por outro membro por algum erro. Ou criam-se duas variáveis diferentes para a mesma função com dois valores diferentes. Quando pensamos no desenvolvimento de trabalhos em conjunto temos que pensar que essas ferramentas colaborativas devem garantir que todo o trabalho que tenha sido realizado não seja perdido por uma publicação indesejada de um dos componentes do grupo e que esteja a disposição sempre um histórico de modificações realizados por todos os integrantes, além de ser possível retornar o sistema à estados anteriores.

Com a evolução das ideias de colaboração em desenvolvimento de software nas décadas de 70 e 80 e a criação de ferramentas e modelos como o CVS - Concurrent Version System (Sistema de Versões Concorrentes) - SVN (Apache Subversion) e GIT (Também um sistema de controle de versões) cada vez mais ficou mais usual encontrar essas ferramentas nos chamados “Ambientes Integrados de Desenvolvimento” (IDEs), que garantem, portanto, a integridade, segurança e controle de modificações do sistema.

Booch e Brown ([BOOCH; BROWN, 2003](#)) afirmam que mecanismos adicionais para colaboração de equipe devem ser trazidos para o ambiente de desenvolvimento para reduzir ainda mais atritos no grupo; eles defendem ambientes de desenvolvimento colaborativo (CDEs) completos - espaços de projetos virtuais em que todos os interessados podem trocar conhecimento, conversar, debater e trabalhar juntos em uma tarefa comum, por exemplo, produtos de software.

## 1.4 Arte digital e colaboração

O meio digital exibe características distintas, que são frequentemente usadas em combinações variadas. É interativo, permitindo formas de navegar, montar ou contribuir

com obras de arte para ir além do evento mental de apenas vivenciá-las. Frequentemente é dinâmico, respondendo a um fluxo de dados em mudança e transmissão de dados em tempo real. A arte nem sempre é colaborativa no sentido original da palavra, mas muitas vezes participativa, baseando-se na entrada de múltiplos usuários (PAUL, 2002).

O trabalho em equipe interdisciplinar tem se tornado cada vez mais comum no desenvolvimento industrial e do conhecimento, mais recentemente no campo da prática de arte em mídia digital, onde a complexidade da tecnologia causou uma mudança da produção individual para a produção em equipe. A complexidade da maioria dos projetos exige diversas formas de conhecimento adquiridas ao longo do tempo por meio da experiência. Para realizar um projeto, é necessário, portanto, que o conhecimento necessário seja fornecido por muitos especialistas, cujos esforços são integrados através de um processo colaborativo. Espera-se também que reunir conhecimento especializado de diversos domínios possa resultar em efeitos sinérgicos, em que o todo se torna algo diferente da soma de suas partes. A integração das abordagens, métodos e estratégias diversificadas dos especialistas idealmente permite uma visão multifacetada de um dado problema e, portanto, aumenta o potencial criativo dos membros da equipe. Essa visão multifacetada só é possível se houver uma compreensão compartilhada do problema e das várias abordagens para resolvê-lo (STEINHEIDER; LEGRADY, 2004).

Parceiros em uma colaboração precisam definir metas mutuamente benéficas para que esses projetos multidisciplinares sejam bem-sucedidos. Os coordenadores fornecem um papel vital nos projetos de produção. Também precisamos reconhecer os diferentes pontos de vista de artistas e design de interação para públicos e usuários (ENGLAND, 2012).

As diferentes visões e perspectivas em colaboração são essenciais para ajudar as pessoas a se entenderem melhor e se inspirarem em criatividade. A criatividade ocorre quando as pessoas são capazes de conectar diferentes referenciais de maneiras que resultam na criação ou descoberta de algo novo. Pense no grande número de novas possibilidades e ideias quando muitas mentes, ou uma combinação de grandes pessoas, são reunidas através de um objetivo compartilhado onde se tem um diálogo sobre determinado contexto ou problema em comum. A capacidade coletiva na solução dos problemas portanto é amplamente aumentada (HARGROVE; SENGE, 1998).

## 1.5 Objetivos

Este trabalho tem por objetivo estudar uma forma eficiente para compartilhar os artefatos gerados pelo Mosaicode e desenvolver ferramentas para permitir a colaboração entre os usuários.

Além do objetivo geral, busca-se alcançar os seguintes objetivos específicos:



- Discutir sobre compartilhamento em arte digital e o auxílio de computadores no trabalho colaborativo.
- Definir os artefatos gerados pelo desenvolvimento no Mosaicode;
- Definir métodos para versionamento dos artefatos;
- Definir como será realizada a distribuição dos artefatos (peer-to-peer ou centralizado);
- Definir como organizar o espaço de trabalho do usuário;
- Criar protótipos de ferramentas definidas pelo estudo realizado dos itens anteriores no ambiente Mosaicode;

## 1.6 Metodologia

O desenvolvimento deste projeto foi dividido em 3 fases e 8 etapas. Estas etapas são apresentadas na Figura 3 e detalhadas neste capítulo.

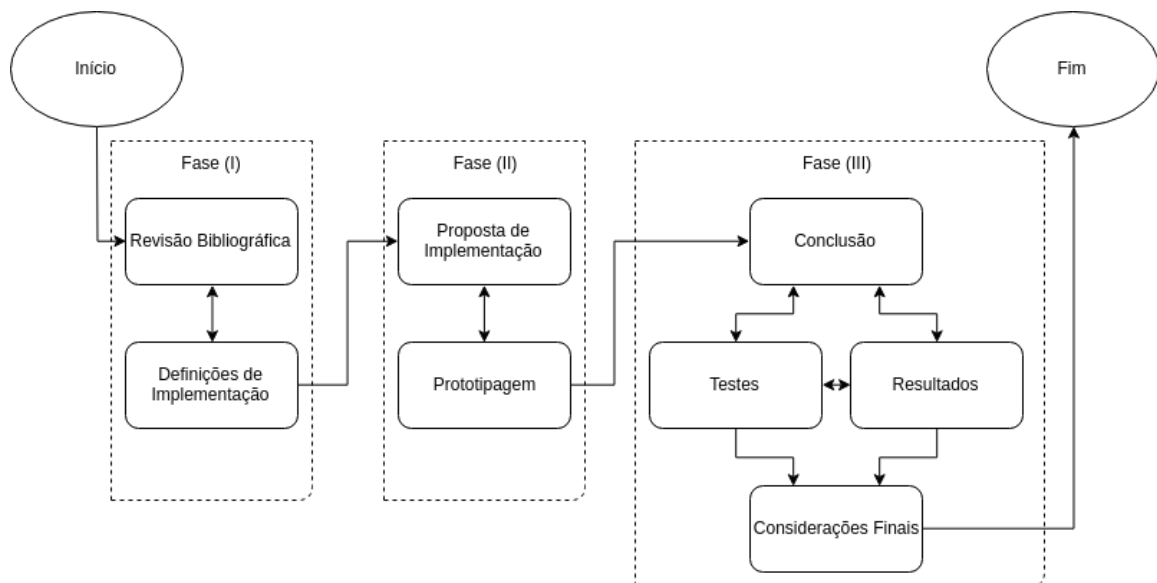


Figura 3 – Fluxograma da metodologia do POC II

- Fase (i): Ao início do projeto foi definido que deveríamos estudar sobre artefatos e métodos de compartilhamento do mesmo. À partir deste ponto, a segunda etapa da fase (i) consiste em definir os artefatos gerados pelo Mosaicode para que possa se determinar também quais serão as modificações a serem feitas para se distribuir os objetos gerados além de estabelecer o modo como será compartilhado cada um desses produtos. As revisões bibliográficas foram feitas sobre o uso de computadores no auxílio à colaboração e trabalho cooperativo. Também foi discutido sobre arte digital

e as interações que podem ser geradas pelos membros de um grupo de arte digital. Foi feita uma pesquisa sobre ferramentas de colaboração nos principais softwares de desenvolvimento em arte digital para se ter uma base do que poderia ser desenvolvido ou melhorado.

- Fase *(ii)*: A segunda fase do trabalho é desenvolver propostas de protótipos e implementação geradas à partir das definições da fase *(i)*. Também serão gerados protótipos baseados nas propostas a fim de observar se os requisitos foram satisfeitos.
- Fase *(iii)*: Ao final da última fase é esperado a implementação das propostas da fase *(ii)*, juntamente com os protótipos gerados, teremos os testes, resultados gerados, conclusão e considerações finais para próximos trabalhos.

## 2 Computer Supported Cooperative Work

CSCW (Computer Supported Cooperative Work), ou trabalho cooperativo auxiliado por computador, é a ciência que estuda as formas de trabalho em colaborativos apoiados por tecnologias. Essas formas de colaboração podem ser desde simples conferências por bate-papos ou aplicativos de comunicação até mesmo sistemas e equipamentos inteiros desenvolvidos para o trabalho em equipe (GRUDIN, 1991).

Os sistemas em geral tendem a ser colaborativos nos dias de hoje, seja para facilitar a comunicação, ou ferramentas para facilitar os processos e coordenação humana, e é esperado um crescimento maior ainda desse tipo de aplicações cooperativas (PENICHERT et al., 2007).

CSCW é dito, também, multidisciplinar uma vez que envolve técnicas de vários campos de estudo das mais diversas áreas, como psicologia, sociologia, antropologia, ciência da computação, economia, ergonomia e etc. em busca de aprimorar as interações entre os usuários de acordo com a necessidade de cada grupo.

### 2.1 Histórico

O termo trabalho cooperativo foi empregado nas ciências sociais ainda no século XIX por economistas como designação de trabalho envolvendo múltiplos participantes. À partir daí viu-se que a troca de informações entre os envolvidos, capacidade de comunicação e o respeito as individualidades de cada um tornavam mais eficaz o trabalho coletivo.

Com o advento do micro-computador em meados da década de 70 e da internet moderna, já na década de 80, surgiu-se a necessidade de estudar como o trabalho colaborativo poderia ser auxiliado por meio das novas tecnologias.

O CSCW é um sub-campo de uma área que é pesquisada, desde os primeiros computadores, de tecnologias a serviço de trabalhos colaborativos. O conceito de CSCW se originou no ano de 1984 num workshop organizado por Irene Greif e Paul Cashman onde pesquisadores de diversas áreas trocaram experiências para desenvolver ideias de como a tecnologia de informação poderia aprimorar os trabalhos em grupo. Os estudos então convergiram para uma ideia comum: “De forma geral, CSCW examina as possibilidades e efeitos do auxílio tecnológico para humanos envolvidos em comunicações de grupos colaborativos e processos de trabalho.” (KOCH; SCHWABE; BRIGGS, 2015).

Ao longo do tempo, diversas pesquisas com diferentes perspectivas foram realizadas para descrever melhor o entendimento sobre o trabalho cooperativo auxiliado por

computador. Hoje, por exemplo, o conceito de comunicação através de computadores no auxílio de um grupo de trabalho é bem difundido e amplamente realizado nos mais diversos seguimentos.

## 2.2 Groupware

O termo groupware foi concebido antes mesmo do CSCW, em 1981 por Peter e Trudy Johnson-Lenz no artigo *“Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium”* (JOHNSON-LENZ; JOHNSON-LENZ, 1981). Sua intenção inicial seria “processos de grupo intencionais, além de software para apoiá-los”<sup>1</sup>.

A diferença entre CSCW e groupware pode ser sutil, mas, devemos ficar atentos ao que cada um desses conceitos se comprometem a realizar. Enquanto CSCW tem foco nas pesquisas acadêmicas sobre o trabalho em grupo auxiliado por computador, o groupware se baseia na implementação dessas ferramentas. Podemos dizer, portanto, que a relação entre CSCW e groupware é que o primeiro diz a respeito do estudo científico, enquanto o segundo descreve a tecnologia (PENICHER et al., 2007).

## 2.3 O ambiente dinâmico do trabalho (Interação)

O trabalho colaborativo tem o potencial para gerar melhores resultados que os trabalhos realizados de forma individual. Isso se dá ao fato de pessoas diferentes terem capacidades e pontos de vistas distintos, podendo serem complementados. Com isso, o grupo pode criar mais alternativas para solução de um determinado problema, e mais ainda, analisar essas alternativas vendo as vantagens e desvantagens de cada uma para tomar a melhor decisão baseado na experiência de cada um (FUKS; RAPOSO; GEROSA, 2002).

Como nos diz Chiavenato:

“Na Era da Informação, lidar com as pessoas deixou de ser um problema e passou a ser a solução para as organizações. Deixou de ser um desafio e passou a ser a vantagem competitiva para as organizações bem sucedidas” (CHIAVENATO, 2008).

---

<sup>1</sup> Original: : *“Intentional group processes plus software to support them”*

## 2.4 Ferramentas

As ferramentas de groupware permitem os indivíduos de um grupo realizarem atividades em conjunto, incluindo correio eletrônico, agendamento de atividades, acesso remoto, compartilhamento de informações, compartilhamento de documentos (arquivos), de código fontes, gerência de fluxo de trabalho, dentre outras (CAMPOS; TEIXEIRA, 2004). Nessa seção discutiremos algumas dessas ferramentas.

Essas ferramentas groupware são embasadas no conceito 3C (Comunicação, coordenação e colaboração) (CAMPOS; TEIXEIRA, 2004).

### 2.4.1 O Modelo 3C

Para que tenhamos um bom planejamento de groupware é necessário aplicarmos alguns conceitos que serão discutidos aqui.

O modelo 3C, apresentado na Figura 4 trata dos três pilares de uma colaboração: Comunicação, cooperação e coordenação. A comunicação trata-se de troca de mensagens, argumentação, negociação. Já a cooperação pelo trabalho conjunto entre as pessoas. A coordenação já se trata do gerenciamento de pessoas.



Figura 4 – Aplicações do Modelo 3C.

Fonte: <<http://tarciomarsouza.blogspot.com/2015/10/modelo-3c-de-colaboracao.html>>

### Comunicação

Talvez seja o mais óbvio, uma vez que para se ter um trabalho em grupo, necessariamente, é preciso haver comunicação entre seus integrantes. Essas ferramentas podem

ser chats (bate-papos), aplicativos de conversação de mensagens rápidas, e-mails, videoconferências, fóruns, etc. Podem ser classificados em síncrona e assíncrona.

Síncrona	Assíncrona
Telefonemas	Chat
Reuniões presenciais	E-mails
Vídeo-conferências	SMS

Tabela 1 – Tabela de Ferramentas de comunicação Síncrona e Assíncrona

Na tabela 1 temos alguns exemplos de tecnologias que auxiliam a comunicação. A comunicação é síncrona, quando se tem resposta imediata, em tempo real e é dita, assíncrona, quando depende da disponibilidade do receptor da mensagem para resposta.

Há diversos propósitos que fazem membros de um grupo se comunicarem. As ferramentas de comunicação assíncrona são utilizadas geralmente quando se deseja a reflexão dos participantes, uma vez que terão mais tempo para que seja tomada suas ações. Já na comunicação síncrona, a velocidade da interação é valorizada, uma vez que o tempo de resposta bem como o tempo de reação dos participantes é instantâneo ou bem curto. Algumas ferramentas voltadas para conversação podem ser desestruturadas, enquanto outras podem ser estruturadas e armazenadas em listas, arvores ou grafos (FUKS et al., 2003).

### Coordenação

Através da comunicação podemos, portanto, gerar compromissos e consequentemente, a necessidade de se gerenciar as garantias de efetivação dos mesmos, bem como o gerenciamento das sequências de ações e tarefas para que essas atividades sejam realizadas. A automação desse processo podemos chamar de *workflow* (fluxo de trabalho). Os softwares de gerenciamento do fluxo de trabalho são conhecidos como WfMS (Workflow Management Systems) ou em português, Sistemas de gerenciamentos de fluxo de trabalho. São sistemas que automatizam todo o processo do manejo de pessoas, atividades, etc. Essas aplicações tendem a ser bastante complexas e flexíveis para que o gerente do *workflow* possa ter um vasto número de alternativas para cada situação.

### Colaboração

Aqui está o intuito de toda nosso trabalho. Primeiro, é importante definir o conceito de equipe: grupo de pessoas que se juntam para alcançar um objetivo comum. E a grande vantagem de se utilizar os computadores para a colaboração entre os membros de uma equipe virtual pode ser feita sem a presença destes (TOWNSEND; DEMARIE; HENDRICKSON, 1998).

As equipes virtuais tem sua formação como qualquer outra equipe não-virtual, de início caótico, até que a maturidade da equipe atinja um equilíbrio de coordenação e

interdependência através do entendimento da mesma como um time. Entretanto, equipes virtuais podem ter pouco ou nenhum contato pessoal, que pode afetar a comunicação e colaboração entre as partes (BEJARANO et al., 2006).

Essa colaboração entre os indivíduos de um grupo podem ocorrer de diversas formas, seja por compartilhamento de informações, arquivos, códigos-fonte, etc. No entanto, para que haja compartilhamento de informações é necessário que tenhamos um espaço compartilhado.

## 2.4.2 Repositórios e Workspace

Repositório, como definição, é onde armazenamos “*coisas*”. No nosso caso, chamaremos essas coisas de artefatos de software. Um repositório, portanto, não só irá guardar as informações deste artefato, mas também compartilhá-la na rede nos casos de trabalho em grupo.

Segundo (ESTUBLIER, 2000), no gerenciamento de um repositório de componentes, há a necessidade de armazenar diferentes componentes de um produto de software e todas as suas versões de forma segura. Isso inclui gerenciamento de versão, modelagem de produto e gerenciamento de objetos complexos.

Esses artefatos, à princípio, podendo ser qualquer coisa, arquivos de texto, imagens, multimídias, ferramentas, códigos, etc, são informações compartilhadas entre as pessoas.

Workspace, num sistemas de arquivos, é onde um software, arquivo ou dado pode ser manipulado ou desenvolvido isolado de outros. É o espaço de trabalho de determinada aplicação. Note-se que uma mesma pessoa pode participar de vários workspaces diferentes. O mesmo vale para grupos distintos.

Algumas características de workspace, segundo (SPELLMAN et al., 1997), devem ter as seguintes características:

- Ser persistente: continue existindo se alguém está “neles” ou não. Isto está em contraste com as ferramentas centradas em reuniões, onde não é possível deixar algo para trás para alguém que pode entrar na interação mais tarde.
- Ser independente de localização: pode ser acessado independentemente da localização do usuário.
- Ter localização transparente: possibilita a interação com qualquer pessoa sem conhecer sua localização física.
- *Stateful* (Dinâmico): Fornecer meios no qual os usuários podem interagir uns com os outros e / ou com documentos selecionados .

### 2.4.3 Ferramentas integradas

Ambientes de desenvolvimento construídos com o intuito de facilitar a criação de novas aplicações através do abrangimento de várias ferramentas em um único software são chamadas de ferramentas integradas.

Em vários sistemas buscamos a integração de ferramentas diferentes para que o usuário tenha disponível tudo o que precisa para desenvolver seu trabalho em uma única solução, diminuindo assim a troca de telas e busca por outras aplicações, aumentando sua produtividade e concentração no trabalho a ser feito.

Esses ambientes normalmente provém quase todo o trabalho que um desenvolvedor necessita, como compiladores, linkers e bibliotecas, depuradores de erro (*debuggers*) e em alguns casos, os IDEs integram ferramentas de controle de código fonte também; eles servem como repositórios compartilhados para os artefatos de código de uma equipe de desenvolvimento de software e também oferecem suporte à colaboração estruturada, permitindo que os desenvolvedores compartilhem, editem e mesquem arquivos de maneira coordenada sem sair do contexto do IDE (HUPFER et al., 2004).

Segundo Booch e Brown (BOOCH; BROWN, 2003), sistemas colaborativos valiosos podem emergir de IDEs com ferramentas aparentemente simples da metodologia 3C (coordenação, comunicação e colaboração).

Segundo (OWOSEN; AKANJI, 2016), os benefícios de se utilizar IDEs são:

- Maximização da produtividade do programador com menos esforço - O ambiente fornece alguns recursos que ajudam os programadores a maximizar suas produtividades. Pouca entrada ou esforço é esperado dos programadores e essa pequena entrada produz produtos maximizados com a ajuda dos recursos contidos na IDE.
- Redução no tempo de desenvolvimento de software - Com as ferramentas de construção de programas disponíveis, o tempo de desenvolvimento fica mais curto. Com essas ferramentas, muitos dos códigos necessários são gerados automaticamente, reduzindo as linhas de código a serem editadas manualmente pelo programador.
- Aplicação dos padrões do projeto ou da empresa - Os padrões podem ser aplicados se o IDE oferecer modelos ou projetos predefinidos ou se as bibliotecas de código forem compartilhadas entre os diferentes membros da equipe que trabalham no mesmo projeto.
- Redução no estresse de desenvolvimento - O estresse transmitido pelos programadores ao desenvolver aplicativos de software complexos pode ser reduzido com a ajuda dos IDEs.



## 3 Colaboração em Arte Digital

A arte digital tem se desenvolvido muito nos últimos tempos, mas esse não é um assunto tão novo quanto podemos imaginar. Sua pré-história começou ainda na década de 1940 (antes mesmo dos computadores), quando o cientista do exército americano, Vannevar Bush, descreveu uma máquina, Memex, em um artigo científico chamado “As We May Think”. Uma mesa com telas translúcidas que permitia aos usuários navegar pelos documentos em várias mídias (do texto à fotografia) e criar sua própria trilha para um corpo de documentação. A máquina ajudaria à mente humana, muita das vezes falha em guardar memórias, a guardar informações através dessas mídias. Apesar de nunca ter sido construída, a Memex nada mais é que o conceito que temos de computadores e seu armazenamento.

A arte digital não se desenvolveu em um isolamento histórico-artístico e incorpora muitas influências de movimentos artísticos anteriores variando de arte conceitual a Fluxus (movimento artístico da década de 60 de cunho libertário, caracterizado pela mescla de diferentes artes), arte postal e experimentos com arte e tecnologia. O ano de 1966 viu a fundação da E.A.T. (Experimentos em Arte e Tecnologia), que nas palavras de seu fundador, Billy Klüver, foi formado por um desejo de “desenvolver uma colaboração efetiva entre engenheiro e artista” (PAUL, 2002). Esse relacionamento entre dois elementos tão distintos, engenheiro e artista foi sempre um dos principais empecilhos no desenvolvimento em conjunto da arte digital.

Hoje temos vários exemplos de colaboração em arte digital. Vão desde orquestras de notebooks à trabalhos inteiros envolvendo interatividade com dança, vídeos e música. Um exemplo disso é o teamLab, em Tokyo, fundado em 2001, é um grupo criativo colaborativo e interdisciplinar que reúne profissionais de várias áreas da prática na sociedade digital: artistas, programadores, engenheiros, animadores Computação gráfica, matemáticos, arquitetos, web designers gráficos e editores gráficos. Referindo-se a si mesmos como “ultra-tecnólogos”, seu objetivo é alcançar um equilíbrio entre arte, ciência, tecnologia e criatividade.

A interatividade também surge da colaboração entre artista e público. Em obras artísticas interativas é possível que o espectador mude o resultado da obra e vivencie experiências imersivas que ajudam tanto o entendimento do mesmo sobre a peça, quanto a do próprio autor.

Segundo (CULTURE, 2015):

“Com obras de arte interativas, as ações e o comportamento do espectador

podem decidir o trabalho artístico em qualquer momento específico. A linha de fronteira entre a obra de arte e o espectador tornou-se mais ambígua. O espectador tornou-se parte da obra de arte em si. Um determinado momento em uma obra de arte é determinado pela presença e pelo comportamento dos espectadores, obscurecendo o limite entre os espectadores e as obras de arte. A obra de arte se torna uma obra de arte incorporando seus visualizadores. Por exemplo, em pinturas anteriores à era digital, as obras de arte são independentes dos espectadores, com um limite claramente definido entre os espectadores e os objetos visualizados. O espectador, como pessoa independente, está sempre enfrentando a obra de arte. Em geral, a pintura permanece sempre a mesma, quer alguém a tenha visto 5 minutos antes, quer alguém esteja ao seu lado ao mesmo tempo.”(CULTURE, 2015)



Figura 5 – Crianças participam de obra de arte digital.

Fonte: <<https://www.digitalmeetsculture.net/article/teamlab-tokyo-based-digital-artist-collaborative/>>

Outro projeto, (LEE et al., 2008), usa controles de vídeo-game para interação. São controles interativos de Nintendo Wii, e criaram-se vários softwares que obedecem a esses controles. Uma de suas aplicações foi o WiiBand, uma banda virtual para que as pessoas possam interagir e criar sons.

Segundo (CONSULTING, 2015), diversos benefícios podem ser desenvolvidos de colaboração artísticas:

- Adquirir conhecimentos artísticos;
- Melhorar a qualidade;
- Melhorar as conexões com artistas;
- Aumentar a participação de artistas;

- Desenvolver e apresentar oportunidades de performance / produção;
- Introduzir um novo programa ou serviço;
- Expandir ou construir programação / serviços ou mudar para novas áreas;
- Para acessar o espaço (por exemplo, ensaio, local de desempenho) ou desenvolver infra-estrutura física;
- Aumentar a diversidade;
- Desenvolver educação e divulgação;

## 3.1 Criação em Arte Digital

A criação de arte digital, geralmente, utiliza-se de softwares de domínio específico para ser realizada. Podemos citar vários exemplos, como o Photoshop, para arte visual e imagens, Audacity para edição de músicas e sons ou o Openshot para criação e edição de vídeos. Embora a maioria desses programas dispõe de interfaces gráficas onde operamos todo o processo de geração da arte, temos também exemplos de softwares onde temos que programar sua execução para obter os artefatos que desejamos. Nesse contexto, podemos citar outros exemplos, como o PureData, Supercollider, Processing e o próprio Mosaiccode. Estes, possuem linguagens de domínio específico para a criação do software, ou seja, linguagens desenvolvidas especificamente para aquela aplicação levando em conta o seu uso. Abaixo discutiremos sobre linguagens de domínio específico e de proposta geral.

### 3.1.1 GPL vs. DSL

Primeiramente, vamos dar nome a cada uma delas. GPL (General-purpose language) ou linguagens de propósito geral, são linguagens que são criadas para fins gerais. Algumas dessas estão há muito tempo no desenvolvimento e são bem consolidadas, como o C, Java, dentre outras linguagens onde podemos criar aplicações dos mais diversos tipos e variadas áreas.

Já as DSL (Domain-specific languages), ou linguagens de domínio específico, são linguagens criadas para determinados tipos de aplicações para melhorar o rendimento tanto do programador quanto da aplicação desenvolvida. Temos, portanto, um pequeno dilema para decisões de projeto, utilizar linguagens de uso geral já consolidadas ou criar uma linguagem nova de domínio específico para melhorar o desempenho e facilidade de programação daquela determinada área?

De acordo com (MERNIK; HEERING; SLOANE, 2005), DSLs oferecem ganhos substanciais em expressividade e facilidade de uso em comparação com linguagens de programação de propósito geral em seu domínio de aplicação. Entretanto, o desenvolvimento

dessas linguagens é trabalhoso, exigindo conhecimento de domínio e desenvolvimento de linguagens.

Algumas vantagens de se utilizar as DSL, segundo (DEURSEN; KLINT, 2002):

- As DSLs podem ser expressadas no próprio idioma de contexto e no nível de abstração do domínio do problema. Portanto, especialistas no tema podem entender, validar, modificar e até mesmo desenvolver programas na linguagem sozinhos.
- Em geral, os programas DSL são concisos, permitem a auto-documentação e em grande parte podem ser reutilizadas para outros propósitos (reuso).
- As DSLs aumentam a produtividade, capacidade de manutenção, confiabilidade e portabilidade.
- As DSLs também podem incorporar na linguagem conhecimentos de domínio e conseqüentemente a conservação e o reuso desse conhecimento no desenvolvimento de aplicações.
- DSLs permitem validação e otimização no nível de abstração do domínio.
- DSLs também melhoram a testabilidade dos sistemas por serem linguagens de alto-nível

## 3.2 Ferramentas para criação de arte digital

Nessa seção abordaremos alguns softwares utilizados para criação de arte digital e o que esses oferecem para suporte para trabalho cooperativo. Daqui também poderemos tirar proveito de algumas ideias interessantes de ferramentas que já existem para colaboração no nosso cenário de arte digital para desenvolvermos uma solução que possa abranger as necessidades dos artistas dentro do contexto do *Mosaiccode*.

- PureData O PureData é uma linguagem de programação visual para desenvolvedores de multimídia. Tem muito em comum com o *Mosaiccode* por também ter um ambiente gráfico com seus componentes já pré-definidos, além de ser também expansível através de plugins.

Em 2011, (TSOUKALAS, 2011) criou um fluxo de trabalho de colaboração utilizando o Pure Data como framework. Algumas dificuldades são encontradas por o Pure Data não possuir ferramentas que permitam a colaboração. Ao fim do artigo, ele sugere algumas funcionalidades para o trabalho colaborativo no PD de acordo com as experiências vividas. Essas são:

- Post-it - uma estrutura query de edição de texto com histórico e possíveis assistencia para imagem/vídeo
  - TODO - Lembrete de informação com avisos de janela flutuante por patch e sub-patch.
  - Structure Chart - Diagramas de sub-patches e abstrações incluídas em um patch e suas abstrações com os rastros de qualquer conexões não existentes entre os patches (como enviar-receber).
  - Controle de Log - Visualização/log de dados minerados de entrada e de qual usuário selecionou as variáveis (por exemplo, para facilitar a investigação e/ou experimentação com controle de parâmetro de padrões de dados de entrada, tempo de reação do performer ou eficiência da mineração de dados.
- Supercollider O SuperCollider é uma plataforma para síntese de áudio e composição através de algoritmos, pode ser usado por músicos, artistas e pesquisadores que trabalham com som. Um dos componentes desta plataforma é uma linguagem de programação musical e um servidor para executar esta linguagem. Também é extensível, ou seja, a ferramenta aceita plugins para funcionalidades. É utilizado principalmente para live-coding, apresentações ao vivo, onde em tempo de execução podemos alterar parâmetros nos algoritmos criando diferentes sons, tornando as performances interativas.

Devido à sua característica de live-coding, temos portanto algumas extensões já prontas para a performance colaborativa (para que mais de uma pessoa possa tocar ao mesmo tempo, como se fosse uma banda).

- TROOP - É um editor/ambiente cooperativo que permite você colaborar simultaneamente a mesma parte de código de máquinas separadas. Além disso é OpenSource e pode ser instalado tanto no python 2 quanto no python 3. No editor, todos compartilham do mesmo código e cada usuário tem uma cor diferente para saber o que cada um está alterando. Todos compartilham o relógio do servidor, já que nesse tipo de performance é de extrema importância a sincronização dos dispositivos (KIRKBRIDE, 2017).
  - SuperCopair - É uma ferramenta que permite a colaboração de código pelo SuperCollider através da nuvem. Utiliza o Pusher <<http://pusher.com>> como provedor do serviço. Compartilham também o mesmo código e pode ser desenvolvido e tocar/parar a música de três maneiras diferentes: somente no seu sistema, somente no sistema de seus amigos ou em todos os sistemas (JUNIOR; LEE; ESSL, 2015).
- Processing

Segundo seu site oficial <<https://www.processing.org/>>, Processing é um “caderno de esboços” (sketchbook) de software flexível além de uma linguagem para se aprender a codificar no contexto das artes visuais. A Processing, desde 2001 trabalha com a alfabetização em software no contexto das artes visuais e também no fluxo contrário, na alfabetização visual dentro do cenário da tecnologia. Dezenas de milhares de estudantes, dsigners, artistas, pesquisadores ou amadores que usam o programa para aprender, se desenvolver e criar protótipos de artes visuais para diversas finalidades.

A princípio não contava com nenhuma ferramenta de colaboração, mas como é um software extensível, a comunidade voltou as atenções para desenvolver uma ferramenta para integrar o trabalho colaborativo dentro da ferramenta. Joel Moniz, integrou a ferramenta ao GitHub através de um plugin que foi chamado de Git Manager <<https://joelmoniz.com/git-manager/>>. Essa aplicação está disponível para ser baixada pelo controlador de plugins do próprio ambiente Processing.

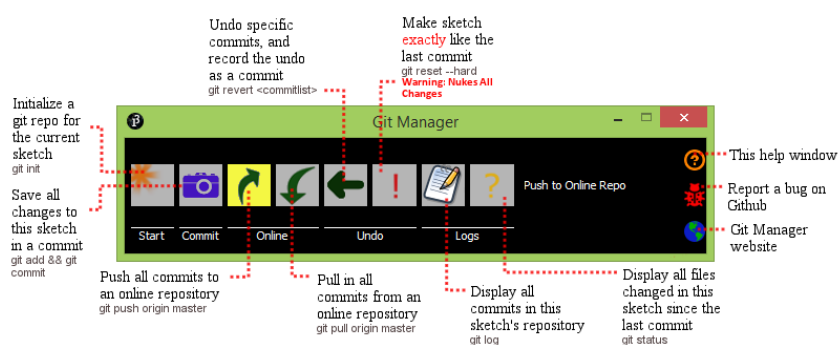


Figura 6 – GitHub Manager para Processing.

Fonte: <<https://github.com/joelmoniz/Git-Manager>>

A Figura 6 apresenta as funcionalidades da última versão da extensão. Suas funcionalidades são as básicas de qualquer repositório Git. Git init (cria repositório), git commit (salva o estado atual para enviar), git push (envia as modificações salvas no commit), git pull (baixa a ultima versão do repositório), git revert (cancela o último envio), git reset (reseta o projeto como o ultimo envio), git log (log de todos os commits) e git status (mostra todos os arquivos modificados desde o último commit).

Com essa ferramenta então, temos integrado no ambiente as facilidades para trabalho em equipe que as plataformas Git oferecem para colaboração.

- Unity

O Unity é uma ferramenta de desenvolvimento gráfico para aplicações de diversas finalidades. Pode ser usado para simuladores, criação de jogos, criação de filmes, projetos arquitetônicos dentre muitos outros. A ferramenta conta com vários objetos, também chamados de Assets. Esses, englobam tanto componentes gráficos, como texturas, scripts e etc. Com esses assets pré-fabricados é possível realizar diversas operações de maneira simples e rápida. São distribuídos pela loja oficial da Unity chamada AssetStore, onde é possível encontrar desde pré-definições pagas à gratuitas. Após a aquisição do componente, é possível importá-lo para seu projeto.

Além disso a Unity também oferece dentro do ambiente o Collab. O manual pode ser achado em sua documentação em <<https://docs.unity3d.com/Manual/UnityCollaborate.html>>. Essa ferramenta permite pequenas equipes salvar, compartilhar e sincronizar projetos num ambiente de nuvem. Foi desenvolvido para dar suporte a times multi-disciplinares (onde há desenvolvedores, artistas, e especialistas de som, video, etc.) para trabalharem em conjunto. Essa ferramenta está ativa do Unity 2017.1 adiante. Os arquivos ficam salvos na nuvem da própria Unity e inclusive pode ser compilado na nuvem com o Cloud Build. Não temos mais detalhes técnicos sobre o desenvolvimento do projeto por ser um software proprietário.

- Mosaicode

O *Mosaicode* é um ambiente de programação visual com o intuito de ajudar artistas no desenvolvimento de aplicações e ainda não possui nenhuma ferramenta para colaboração desenvolvido.

## 4 Prototipação

Os protótipos das ferramentas selecionadas serão discutidas nesse capítulo. Alguns conceitos importantes para o desenvolvimento dos mesmos serão também discutidos aqui. Precisamos agora definir quais ferramentas serão implementadas para que haja uma colaboração de forma eficiente. Para que haja uma melhor comunicação entre os usuários, vamos integrar ao sistema um chat. Como já foi discutido anteriormente na seção 2.4.3 a integração das ferramentas aumenta a produtividade dos desenvolvedores. Devemos também permitir o compartilhamento dos artefatos gerados pelo sistema. Note que para que isso seja possível de forma coesa, deveremos fazer algumas mudanças no que se diz respeito ao formato atual dos arquivos do sistema.

### 4.1 Decisões de projeto

Foi escolhido de acordo com as revisões sobre CSCW aplicações para comunicação e compartilhamento, como já vimos, ferramentas aparentemente simples com o conceito do modelo 3C aplicado, podem se tornar poderosas soluções para o trabalho colaborativo.

As seguintes decisões de projeto influenciaram o desenvolvimento deste trabalho.

#### Chat

O chat, como na proposta inicial foi sugerido, será um bate-papo em grupo de rede local. As conexões via rede local permitem o multicasting de dados, que será importante para reconhecermos quem estará com o sistema aberto à conversação.

#### Multicast

Existem três modos de endereçamento de dados em redes:

- Broadcast - Envia a mensagem para todos os usuários da rede.
- Unicast - Envia a mensagem para um nó específico da rede.
- Multicast - Cria-se um grupo de mensagem de determinados indivíduos da rede.

Como a proposta do chat em grupo é para que as pessoas se comuniquem dentro do Mosaicode, a criação de um grupo de pessoas que utilizarão o serviço é a opção mais atraente. Deste modo apenas as pessoas que estarão no grupo de mensagens definido para o Mosaicode na rede interna receberão os dados do sistema de comunicação.



Para ter maior privacidade e praticidade, optamos por implementar um chat que possa ser *peer-to-peer* (pessoa para pessoa) e todos para todos (um canal multicast onde todos podem se comunicar). Essas propostas também auxiliam no contato via rede local e não necessitam de nenhum servidor para que aconteça toda a comunicação.

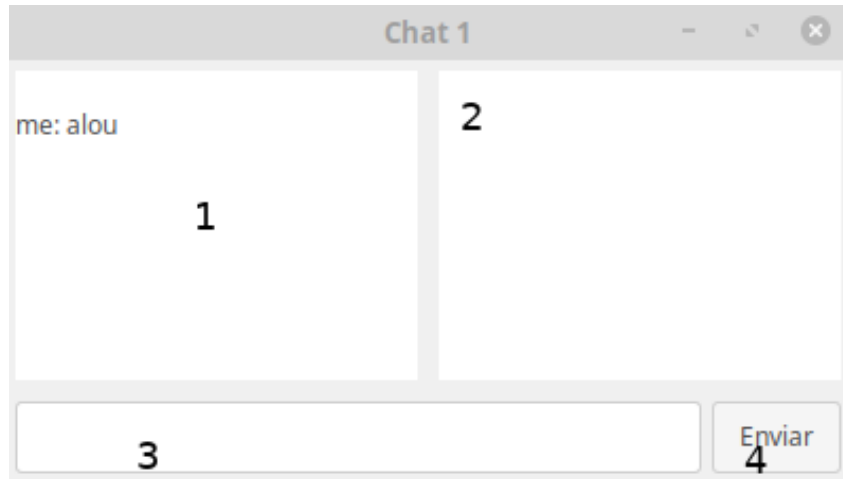


Figura 7 – Tela de chat do protótipo desenvolvido

A Figura 7 representa a tela de chat criada. Em 1, temos a tela com o histórico de conversas do chat. Em 2 temos a lista de pessoas conectadas, as quais poderemos mandar mensagens. Em 3 o campo de entrada do chat e em 4 o botão de enviar a mensagem para a conversa corrente.

## Envio de arquivos

O envio de arquivos também é uma funcionalidade desejada para alcançar a cooperação e colaboração no ambiente Mosaiccode. Podemos escolher um usuário e um arquivo, e enviar via uma conexão entre os dois elementos o arquivo XML podendo ser um bloco ou um diagrama. O usuário receptor deverá aceitar ou não o arquivo para download. Ficará o arquivo então em uma pasta chamada “Received” quando for baixado.

## Workspace compartilhado

Teremos também um workspace compartilhado. Uma pasta “Shared” em cada uma das aplicações enviará por multicast os arquivos contidos nela e cada um dos usuários poderá importá-los a hora que quiserem ao seu projeto. O download do bloco/diagrama também irá para a pasta “Received” do usuário que baixou o arquivo.

O usuário também poderá atualizar a sua lista de arquivos compartilhados (o que fará enviá-la novamente a todos os integrantes daquele canal multicast) ou também pedir que todos o mandem seus arquivos atualizados da pasta compartilhada.

## 4.2 Resultados e implementação

Para a implementação dos protótipos tivemos que criar protocolos de rede para a comunicação local dos usuários. Foram criados portanto dois protocolos, um multicast, para fim de comunicação em grupo e um unicast para fim de comunicação peer-to-peer.

Todas as APIs para desenvolvimento do projeto serão baseadas nas APIs padrões do Python 2.7 e as que já estão incorporadas ao Mosaiccode, como a GTK 3.0 (biblioteca gráfica) e BeautifulSoup (utilitário para resolução de arquivos XML). Ao final, quando estiver com todos os requisitos apresentados nas propostas de mudanças de bloco e workspace apresentados neste trabalho, será de fácil importação o plugin para o sistema.

### Diagramas

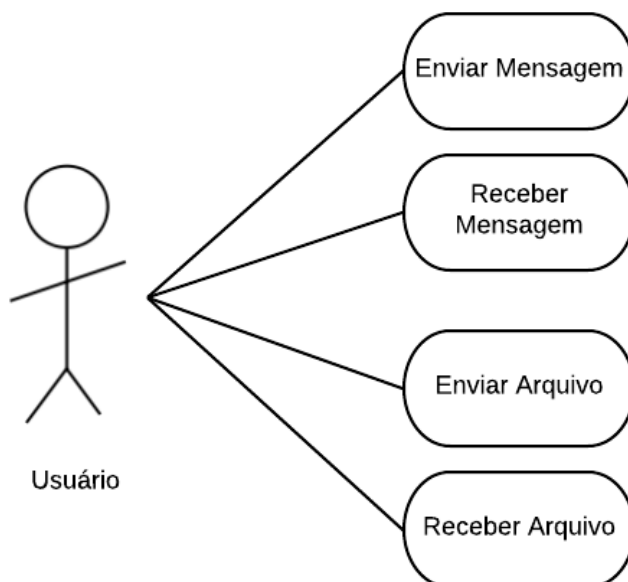


Figura 8 – Caso de uso: peer-to-peer

Os requisitos para o caso de uso de um único usuário para outro usuário específico pode ser visto na figura 8 em que um indivíduo pode enviar uma mensagem de chat para outro ou um arquivo.

Já o diagrama de caso de uso para o grupo é especificado na Figura 9 em que um usuário pode enviar mensagem de chata para o grupo, enviar a lista de arquivos da pasta *Shared*, pedir a lista de arquivos de todos os usuários da pasta compartilhada

As mensagens do protocolo multicast como pode ser visto na figura 10 são as seguintes:

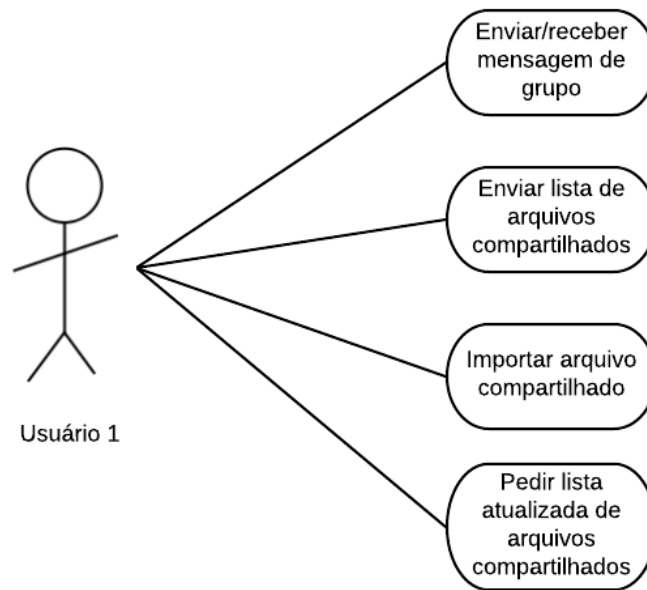


Figura 9 – Caso de uso: multicast

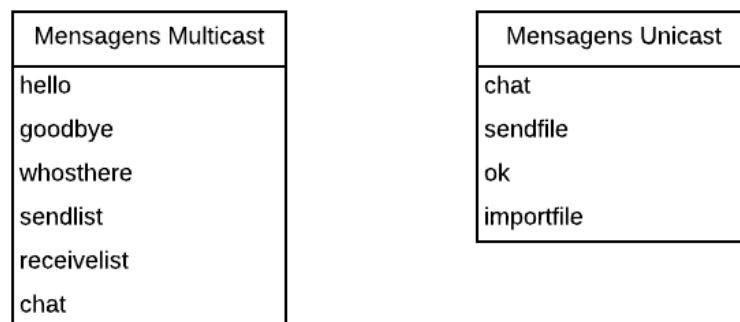


Figura 10 – Mensagens multicast e unicast

- hello - Mensagem utilizada para ser enviada para se adicionar a lista de amigos dos outros usuários quanto como heartbeat (mensagem para dizer que ainda está ativo no grupo).
- goodbye - Mensagem utilizada para dizer que não está mais no grupo e em seguida ser removido da lista de usuários dos outros membros.
- whosthere - Mensagem para perguntar quem está ativo na rede para ser adicionada a sua lista de usuários ativos no grupo. É esperado depois dela a mensagem de hello por todos do grupo.
- sendlist - Envia a lista de arquivos contidos na pasta Shared do usuário.
- receivelist - Pede para todos os usuários a lista atualizada da pasta Shared deles.

- chat - Mensagem de chat em grupo. Essa mensagem será identificada como uma mensagem de chat para ser reproduzida na aba de chat em grupo.

As mensagens do protocolo unicast, figura 10 são:

- chat - Mensagem de chat individual. Essa mensagem será identificada como uma mensagem de chat para ser reproduzida na aba de chat referente ao usuário.
- sendfile - Mensagem para enviar um arquivo. Espera-se uma mensagem de ok do outro usuário para iniciar uma conexão entre os dois para a transferência do arquivo.
- ok - Mensagem de confirmação para realização de uma transferência de dados entre dois usuários.
- importfile - Mensagem de importação de um arquivo da lista de arquivos compartilhados na pasta Shared de um usuário. É esperado um ok da outra pessoa para se realizar a transferência.

### 4.2.1 Protocolos detalhados

Protocolos de rede são conjunto de regras de comunicação entre dois computadores. Definimos os protocolos de comunicação para que possa haver padrão na conversa entre dois indivíduos. Imagine que você faça uma pergunta em português e o ouvinte dela te responda em alemão! Não é esperado isso, portanto, seria impossível (a menos que você saiba alemão) estabelecer a comunicação entre os dois indivíduos. O mesmo acontece com os protocolos de rede.

Em uma conversação entre duas pessoas também temos regras. Um exemplo pode ser a seguinte conversa:

-“Olá, tudo bem”?

- “Tudo bem, e você”?

Veja que o interlocutor responde a pergunta como o esperado. Significa que a pergunta foi recebida com sucesso, interpretada e houve uma resposta a fim de se estabelecer uma conversação. Este exemplo também acontece nos protocolos de rede. Para se haver a conexão entre duas máquinas é trocada uma série de mensagens e elas devem ter um determinado formato. Nessa seção veremos os padrões utilizados para os protocolos desenvolvidos nesse trabalho para a comunicação de rede.

Na Figura 11 temos o esquema do protocolo desenvolvido. O cabeçalho do protocolo consiste em 6 itens:

- Nome - O nome do protocolo. No nosso caso, o nome será: MCSCWP (Mosaicode Computer Supporting Collaborate Working Protocol)

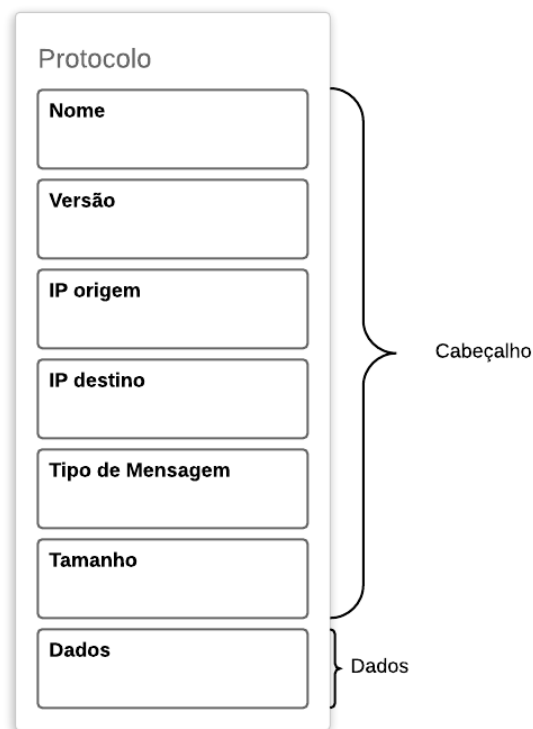


Figura 11 – Protocolo de comunicação proposto e desenvolvido neste trabalho

- Versão - A versão do protocolo que está sendo utilizada. No nosso desenvolvimento estamos na versão 1.0
- IP origem: O IP e porta do computador que está enviando a mensagem.
- IP destino: O IP e porta do computador que está receberá a mensagem
- Tipo de Mensagem: O tipo de mensagem que está sendo enviada: hello, goodbye, etc. As mensagens podem ser vistas na figura 10
- Tamanho: O número de bytes que serão utilizados nos dados.

Já os dados podem ser coisas diferentes, dependendo do tipo de mensagem:

- hello - Nome do usuário.
- goodbye - Nome do usuário.
- chat - Mensagem que foi enviada.
- sendlist: A lista de arquivos em formato XML.
- sendfile: O arquivo a ser enviado.
- importfile: arquivo ou caminho do arquivo a ser importado

Note que todas as mensagens do protocolo serão resolvidas pela aplicação uma vez que o mesmo foi desenvolvido na camada de aplicação de rede.

## 4.2.2 Aplicação

A aplicação deverá ser capaz de identificar as mensagens, armazenar e armazenar os dados de todos os usuários do grupo multicast.

Para isso, temos uma lista de uma estrutura de dados onde cada usuário terá salvo:

- IP - IP de cada usuário.
- Nome - Nome definido por cada usuário.
- ListaArquivos - XML com os arquivos contidos na pasta Shared.

Além disso, cada mensagem fará o sistema responder de formas diferentes.

Ao entrar no sistema, você tem que definir seu nome de usuário. À partir daí, é enviado ao grupo multicast as mensagens: hello, whosthere, sendlist e receiveList. Essas mensagens serão capazes de atualizar a lista de usuários de todos da rede com o seu nome, IP e lista de arquivos compartilhadas do novo membro do grupo. Além de o novo membro do grupo ser também atualizado com os dados de todos que já estavam previamente conectados ao sistema. As figuras 12 e 13 representam as ações tomadas pela aplicação e o envio das mensagens no caso de uma entrada ao sistema.

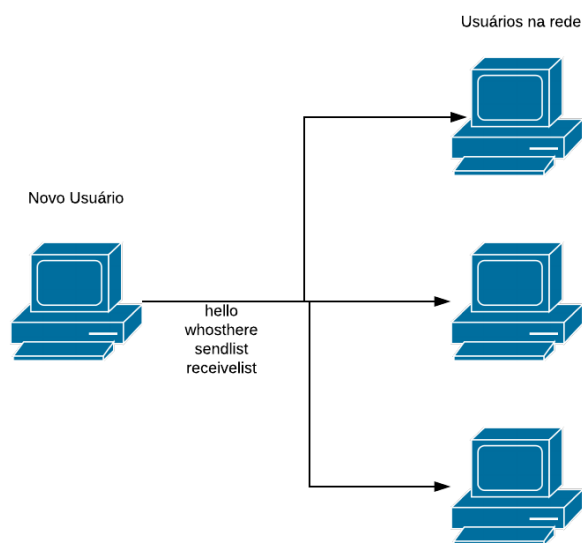


Figura 12 – Mensagens enviadas pelo sistema na entrada de um novo membro no grupo

As mensagens de goodbye farão os sistemas atualizarem seus usuários removendo aquele que enviou a mensagem. Também serão removidos usuários em que o TTL (time

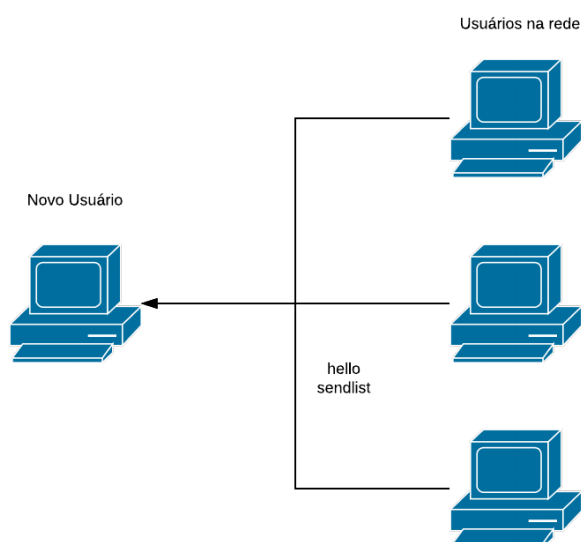


Figura 13 – Resposta do grupo à entrada no sistema da figura 12

to live) do heartbeat - mensagem que de tempos em tempos envia na rede o hello para saber que está ativo ainda - expirar a 300 segundos.

O usuário também poderá enviar sua lista atualizada e baixar as listas atualizadas a qualquer momento no botão "Atualizar Listas" contido no sistema.

## 5 Conclusão

O trabalho propôs o estudo de formas de compartilhamento de artefatos gerados pelo Mosaicode e o desenvolvimento de ferramentas para que a colaboração fosse efetiva entre os usuários. Entretanto, os artefatos gerados pelo mesmo não são usuais e não se tem uma regra geral de colaboração do mesmo como em outras ferramentas já estabelecidas como desenvolvimento em conjunto com o GitHub.

Para isso, na primeira fase do trabalho, foi necessário bastante estudo sobre o que é trabalho colaborativo em computadores e como podemos utilizar essas técnicas em sistemas voltados para a arte digital. Vimos alguns exemplos de aplicações já conhecidas no mercado de arte digital e suas ferramentas de colaboração. Nesse estudo, percebemos que pequenas ferramentas com o conceito do 3C (comunicação, colaboração e coordenação) poderiam resolver o problema proposto.

Na segunda etapa do trabalho, foi definido portanto o que deveríamos compartilhar de forma a realizar o trabalho em conjunto bem como modificações a serem implementadas no formato atual do sistema para que esse compartilhamento fosse efetivo. Foi então desenvolvida propostas também de ferramentas para que a colaboração fosse permitida. Nessa parte foi definido nosso workspace (espaço de trabalho) com as pastas Shared e Received e a proposta de ferramentas de comunicação peer-to-peer e em grupo, sem a necessidade de se utilizar um servidor centralizado.

Na última etapa do trabalho proposto então foi definido um protocolo de rede no Mosaicode para que a comunicação entre os usuários, bem como o compartilhamento de arquivos entre eles fosse feita. Também foi desenvolvido protótipos e soluções para que a ferramenta pudesse ser criada e utilizada futuramente no sistema.

Com isso alcançamos nossos objetivos seguindo a metodologia estipulada no início deste trabalho e o estudo sobre compartilhamento em arte digital em nosso estudo de caso, o sistema Mosaicode.

### Trabalhos futuros

As propostas de modificação das tags de XML ainda estão sendo analisadas para a implementação das ferramentas criadas aqui ao Mosaicode. Ao serem implementadas será necessário criar um sistema de gerenciador de versionamento dos artefatos, para que os blocos e diagramas possam ser identificados como diferentes apesar do mesmo nome.

Como trabalhos futuros pretende-se expandir a ferramenta tendo em vista abranger um número maior de pessoas. A criação de um servidor e um repositório online, para



que pessoas possam compartilhar com o mundo suas criações seria um trabalho incrível. Também pode-se anexar mais mensagens no protocolo como criar grupos de trabalho e workspaces compartilhados em um servidor local. Outro ponto interessante seria anexar ferramentas de controle à ferramenta, pois as desenvolvidas aqui se referiram a comunicação e colaboração.

Na versão atual do *Mosaicodetemos* os blocos como XML. O XML como é uma linguagem de marcação, pode ser adicionada tags em seu corpo para definirmos suas propriedades, portanto, sugerimos a utilização das novas tags a fim de não haver conflito na importação de novos blocos:

- `<author>` - O autor do bloco. Deverá conter o nome de usuário do bloco.
- `<ver>` - Versão do bloco. Se um bloco é atualizado, sua versão deverá incrementar para controle do mesmo.
- `<date>` - Data e hora de publicação do bloco.
- `<last-modified>` - A data e hora da última atualização do bloco.
- `<sub-author>` - O usuário que fez a última modificação publicada.

# Referências

- AHMED S.U., J.; TRIFONOVA, A.; SINDRE, G. *Conceptual framework for the intersection of software and art*. [S.l.: s.n.], 2009. Citado na página 11.
- BEJARANO, V. C. et al. Equipes virtuais—um estudo de caso na indústria têxtil norte-americana. *Production*, SciELO Brasil, v. 16, n. 1, p. 161–170, 2006. Citado na página 22.
- BOOCH, G.; BROWN, A. Collaborative development environments. *Advances in Computers*, v. 59, p. 1–27, 01 2003. Citado 2 vezes nas páginas 14 e 23.
- CAMPOS, E.; TEIXEIRA, F. L. C. Adotando a tecnologia de informação: análise da implementação de sistemas de "groupware". *SciELO Brasil*, 2004. Citado na página 20.
- CHIAVENATO, I. *Gestão de pessoas*. [S.l.]: Elsevier Brasil, 2008. Citado na página 19.
- CONSULTING, P. G. The art of collaboration: A toolkit for nonprofit arts organizations. *PACIFIC OPERA VICTORIA*, 2015. Citado na página 25.
- CULTURE, D. M. *About teamLab, Tokyo-based digital artist collaborative*. 2015. Disponível em: <<https://www.digitalmeetsculture.net/article/teamlab-tokyo-based-digital-artist-collaborative/>>. Citado 2 vezes nas páginas 24 e 25.
- DEURSEN, A. V.; KLINT, P. Domain-specific language design requires feature descriptions. *CIT. Journal of computing and information technology*, SRCE-Sveučilišni računski centar, 2002. Citado na página 27.
- ENGLAND, D. Digital art and interaction: Lessons in collaboration. In: *CHI '12 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2012. (CHI EA '12), p. 703–712. ISBN 978-1-4503-1016-1. Disponível em: <<http://doi.acm.org/10.1145/2212776.2212841>>. Citado na página 15.
- ESTUBLIER, J. Software configuration management: A roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA: ACM, 2000. (ICSE '00), p. 279–289. ISBN 1-58113-253-0. Disponível em: <<http://doi.acm.org/10.1145/336512.336576>>. Citado na página 22.
- FUKS, H.; RAPOSO, A. B.; GEROSA, M. A. Engenharia de groupware: desenvolvimento de aplicações colaborativas. In: *XXI Jornada de Atualização em Informática, Anais do XXII Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2002. v. 2, p. 89–128. Citado 2 vezes nas páginas 13 e 19.
- FUKS, H. et al. Do modelo de colaboração 3c à engenharia de groupware. *Simpósio Brasileiro de Sistemas Multimídia e Web-Webmídia*, p. 0–8, 2003. Citado na página 21.
- GONÇALVES, L. L. *Sound Design com o Mosaicode*. Dissertação (Monografia (Bacharelado em Ciência da Computação)) — Universidade Federal de São João del-Rei, 2017. Citado 2 vezes nas páginas 11 e 12.

- GRUDIN, J. Cscw. *Commun. ACM*, ACM, New York, NY, USA, v. 34, n. 12, p. 30–34, dez. 1991. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/125319.125320>>. Citado na página 18.
- HARGROVE, R. A.; SENGE, P. M. *Mastering the art of creative collaboration*. [S.l.]: McGraw-Hill New York, 1998. Citado na página 15.
- HUPFER, S. et al. Introducing collaboration into an application development environment. In: *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 2004. (CSCW '04), p. 21–24. ISBN 1-58113-810-5. Disponível em: <<http://doi.acm.org/10.1145/1031607.1031611>>. Citado na página 23.
- JOHNSON-LENZ, P.; JOHNSON-LENZ, T. Consider the groupware: Design and group process impacts on communication in the electronic medium. *Studies of Computer-Mediated Communications Systems: A Synthesis of the Findings*, research report 16, Computerized Conferencing and Communications Center, New ... , v. 16, 1981. Citado na página 19.
- JUNIOR, A. D. de C.; LEE, S. W.; ESSL, G. Supercopair: Collaborative live coding on supercollider through the cloud. In: *International Conference on Live Coding*. [S.l.: s.n.], 2015. Citado na página 28.
- KIRKBRIDE, R. Troop: A collaborative tool for live coding. In: *Proceedings of the 14th Sound and Music Computing Conference*. [S.l.: s.n.], 2017. p. 104–9. Citado na página 28.
- KOCH, M.; SCHWABE, G.; BRIGGS, R. O. Cscw and social computing. *Business & Information Systems Engineering*, v. 57, n. 3, p. 149–153, Jun 2015. ISSN 1867-0202. Disponível em: <<https://doi.org/10.1007/s12599-015-0376-2>>. Citado na página 18.
- LEE, H. J. et al. Wiiarts: creating collaborative art experience with wiiremote interaction. In: . [S.l.: s.n.], 2008. p. 33–36. Citado na página 25.
- MERNIK, M.; HEERING, J.; SLOANE, A. M. When and how to develop domain-specific languages. *ACM computing surveys (CSUR)*, ACM, 2005. Citado na página 26.
- OWOSEN, A. T.; AKANJI, S. Survey on adverse effect of sophisticated integrated development environments on beginning programmers' skillfulness. *International Journal of Modern Education and Computer Science*, Modern Education and Computer Science Press, v. 8, n. 9, p. 28, 2016. Citado na página 23.
- PAUL, C. Renderings of digital art. *Leonardo*, MIT Press, v. 35, n. 5, p. 471–484, 2002. Citado 2 vezes nas páginas 15 e 24.
- PENICHER, V. M. R. et al. A classification method for cscw systems. *Electronic Notes in Theoretical Computer Science*, Elsevier, v. 168, p. 237–247, 2007. Citado 2 vezes nas páginas 18 e 19.
- SPELLMAN, P. J. et al. Collaborative virtual workspace. In: *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: The Integration Challenge*. New York, NY, USA: ACM, 1997. (GROUP '97), p. 197–203. ISBN 0-89791-897-5. Disponível em: <<http://doi.acm.org/10.1145/266838.266902>>. Citado na página 22.

STEINHEIDER, B.; LEGRADY, G. Interdisciplinary collaboration in digital media arts: A psychological perspective on the production process. *Leonardo*, MIT Press, v. 37, n. 4, p. 315–321, 2004. Citado na página 15.

TOWNSEND, A. M.; DEMARIE, S. M.; HENDRICKSON, A. R. Virtual teams: Technology and the workplace of the future. *Academy of Management Perspectives*, Academy of Management Briarcliff Manor, NY 10510, v. 12, n. 3, p. 17–29, 1998. Citado na página 21.

TSOUKALAS, K. A collaboration workflow from sound-based composition to performance of electroacoustic music using «pure data» as a framework. Bauhaus Universität Weimer, 2011. Citado na página 27.