

# (Orchidea) Uma primeira aplicação para práticas musicais coletivas na Orquestra de Ideias

Flávio Luiz Schiavoni, Eduardo Xavier da Silva,  
Paulo Gabriel Nunes Cançado<sup>1</sup>

<sup>1</sup>Departamento de Computação - DCOMP  
Universidade Federal de São João del-Rei - UFSJ  
São João del-Rei - MG - Brazil

fls@ufsj.edu.br, eduardo\_xavier\_silva@hotmail.com

paulo.g\_cancado@yahoo.com.br

**Abstract.** *The collective musical practice by the means of computing devices is popular, as it can be seen in the Laptop Orchestras. This paper presents the development of an application to create the Orchidea, an orchestra of ideas, that intends to help musicians to play cooperatively or collaboratively using the cellphone as a musical instrument. The architecture of a first application is presented in this paper as a layered architecture and each layer is decomposed into several components. This paper also presents some project decision and the implementation of the application.*

**Resumo.** *A prática musical coletiva por meio de dispositivos computacionais já é popular, como é o caso das Orquestras de computadores portáteis. Este artigo apresenta o desenvolvimento de uma aplicação para a criação da Orchidea, uma orquestra de ideias, que pretende auxiliar a prática musical colaborativa utilizando celulares como instrumento. A arquitetura de uma primeira aplicação é apresentada neste artigo bem como uma arquitetura em camadas e cada camada é decomposta em seus diversos componentes. Este artigo traz ainda as decisões de projeto e implementação desta aplicação.*

## 1. Introdução

A prática musical em conjunto por meio de dispositivos computacionais já é popular, como é o caso das Orquestras de computadores portáteis (“Laptop orchestra” - LOrc) [Wang et al. 2009]. Estas orquestras têm permitido a pesquisa e criação de novos instrumentos em diversas Universidades no mundo em projetos de pesquisa que transitam na fronteira interdisciplinar entre computação, engenharia e música. Com a evolução da tecnologia móvel, os aparelhos celulares e “*smartphones*” se tornaram computadores portáteis presentes na vida cotidiana de muitas pessoas. O objetivo inicial destes dispositivos era a comunicação e hoje os mesmos cumprem este papel ultrapassando a simples comunicação de voz e permitindo a comunicação por outras mídias como textos, fotos, documentos e vídeos. A expansão de tais limites comunicacionais foi possível graças à evolução da capacidade dos dispositivos principalmente no que tange a conectividade, a capacidade de processamento aliada a possibilidade de manipulação de materiais multimídia e sua capacidade de armazenamento.

A partir desta evolução surgiu a possibilidade de implementação de aplicativos que transformam dispositivos móveis e ubíquos em instrumentos musicais permitindo, por meio desta convergência, o surgimento da área de pesquisa chamada Música Ubíqua (ou Ubiquitous Music - Ubimus) [Keller et al. 2014]. Dentro do ecossistema das práticas musicais em dispositivos ubíquos, vários aplicativos musicais surgiram e encontram-se disponíveis para dispositivos móveis sendo, porém, que a maioria dos mesmos só oferece suporte a interação individual ou são focados para serem utilizados por um único usuário.

A utilização de aparelhos celulares para práticas musicais é vista como a evolução das orquestras de computadores portáteis (*Laptop orchestras*) devido à simplicidade, capacidade e praticidade de tais dispositivos. Estas orquestras serão apresentadas na Seção 2 deste documento. No entanto, há uma carência de pesquisa e desenvolvimento de aplicativos que permitam a interação de músicos por meio de dispositivos móveis em tempo real.

Diante deste cenário, este trabalho apresenta a criação de uma primeira ferramenta para a **Orchidea** (“Orchestra of ideas”) uma grupo que pretende promover a orquestração de dispositivos móveis para práticas musicais interativas.

## 2. Orquestras Digitais

A incorporação dos computadores ao ambiente musical ocorreu somente após a quarta geração de computadores, marcada pelos microprocessadores e computadores pessoais devido ao barateamento dos componentes do mesmo. Isso possibilitou que os computadores pessoais fossem amplamente difundidos no mundo, inclusive no ambiente musical. O surgimento da capacidade computacional para a criação musical na segunda metade do século XX [Mathews 1963] despertou o interesse para a criação de música colaborativa que utiliza redes de computadores para o intercâmbio de informações entre músicos. A música, como prática social, humana e cultural costuma ser uma atividade coletiva e a incorporação dos computadores para práticas musicais não alterou este cenário.

Nesse contexto, surge em 1978 o grupo musical “*The League of Automatic Music Composers*”. Precusores na incorporação de computadores pessoais e circuitos eletrônicos na música, o grupo musical atuou na Bahia de São Francisco difundindo esse novo método de criação musical.

Posteriormente, iniciou-se a iniciativa de criação de Orquestras de Laptop (*LOrc*) principalmente no meio acadêmico. Uma das primeiras orquestras de computadores foi criada em 2005 por Perry Cook e Dan Trueman utilizando um conjunto de quinze meta-instrumentos baseados em computadores portáteis. Esta orquestra, conhecida como “Princeton Laptop Orchestra” ou simplesmente PLOrk [Trueman 2007], surge com o intuito de explorar o papel da tecnologia na música colaborativa além de suplantando as dificuldades intrínsecas do emparelhamento de *laptops*.

Dan Trueman compara uma orquestra tradicional com sua orquestra de *laptops* e afirma que, mesmo a orquestra de *laptops* sendo um tanto quanto diferente, seu objetivo não se difere em nada de uma orquestra tradicional no que tange a capacidade musical alcançada por ambas. Um concerto da PLOrk pode ser visto na Figura 1.

Este modelo de orquestra digital foi amplamente difundido no meio universitário por ser uma oportunidade de integrar, em um mesmo projeto, cientistas da computação,



**Figura 1. Concerto da Orquestra de Laptops de Princeton (PLOrk).**



**Figura 2. Orquestra de celulares de Stanford.**

engenheiros, técnicos e músicos. Por esta razão, diferentes Universidades possuem uma prática coletiva similar onde busca-se alcançar a interdisciplinaridade entre música e tecnologia, como por exemplo: “Stanford Laptop Orchestra” [Wang et al. 2009] (SLOrk), “The Carnegie Mellon Laptop Orchestra” [Dannenberg et al. 2007] (CMLO), “Linux Laptop Orchestra” [Bukvic et al. 2010] (L<sup>2</sup>Ork) (Virginia Tech), “Collaborative Improvisation and Laptop Ensembles” [Lee et al. 2011] (LOLC), e “Concordia Laptop Orchestra” [Tsbary 2014] (CLOC), entre outras. É importante notar que em várias destas Universidades a LOrc é utilizada para integrar atividades de pesquisa, ensino e extensão.

Estes grupos possuem em comum a característica de explorar a tecnologia de maneira diferente tentando extrair o máximo de possibilidades de seus computadores pessoais sendo uma prática comum integrar a seus laptops sensores, comunicação em rede e atuadores. Além disto, algumas destas orquestras também iniciaram pesquisas para a utilização de “*smartphones*” como meta-instrumentos musicais.

Estes dispositivos móveis, apesar de terem menor capacidade de processamento que *laptops*, possuem diferentes interfaces com o usuário devido a quantidade de sensores que os mesmos possuem. Assim, é possível que o músico utilize em sua performance musical não apenas o teclado e mouse como interfaces para interação mas câmeras, sensores de proximidade, acelerômetros, magnetômetros, telas sensíveis ao toque, microfone, sensor de brilho, giroscópios e demais sensores [Essl and Rohs 2007] [Keefe and Essl 2011] [Tanaka 2004] [Roberts et al. 2013].

Devido a estes sensores e processamento, a utilização de telefones móveis como instrumentos musicais também é uma prática já existente. Um dos pioneiros na área de desenvolvimento de instrumentos utilizando telefones celular foi Greg Schiemer [Schiemer and Havryliv 2005] em seu Pocket Gamelan. Nesta performance, o citado autor utilizou o *bluetooth* dos dispositivos móveis para a interação entre dispositivos e a ligação *wireless* como mecanismo para o envio da musica que está sendo executada.

Há atualmente algumas orquestras digitais baseadas em tecnologias móveis, como a “Mobile Phone Orchestra” da Universidade de Stanford [Oh et al. 2010], apresentada na Figura 2, e compor peças musicais para a execução exclusiva nesta plataforma tem se tornado um desafio para compositores [Wang et al. 2008].

### 3. Proposta da Orchidea

A Orchidea é uma proposta de criação de um grupo artístico / musical que utiliza tecnologias acessíveis para suas práticas artísticas na Universidade Federal de São João del-Rei. O grupo pretende permitir que pessoas interessadas em práticas musicais assistidas por computador possam se reunir para a composição e performance musical. Entre nossos objetivos está permitir a criação de música colaborativa por um conjunto de pessoas, sendo estas músicos ou não músicos, por meio de seus dispositivos cotidianos. Utilizando a música para tal interação, cada participante poderá se conectar à orquestra com um instrumento e trocar dados musicais em tempo real de maneira a permitir a criação coletiva.

Este grupo tem um intuito de poder levar a criação sonora colaborativa para qualquer lugar fazendo uso de tecnologias ubíquas como a estrutura de uma rede local. Um requisito desta proposta é que os protocolos de comunicação das aplicações sejam o mais aberto e compatível possível com aplicativos musicais já existentes em algumas plataformas. Isto permitirá que outros instrumentos como notebooks e computadores *desktop* possam fazer parte da orquestra ampliando com isto as possibilidades estéticas.

Também é importante que as ferramentas desenvolvidas sejam elaboradas de maneira que quaisquer mudanças que virem a ser necessárias sejam feitas de forma bem simples. Outro requisito listado inicialmente é que as ferramentas possam garantir a prática individual com o instrumento digital, sem a necessidade de haver demais usuários conectados.

Com isto, podemos enumerar os seguintes requisitos de nosso grupo:

- permitir que os participantes troquem materiais sonoros no caso de estarem tocando em redes,
- permitir que os usuários utilizem instrumentos distintos,
- que as ferramentas sejam abertas e utilizem tecnologias livres,
- que as arquiteturas permitam modificações nas ferramentas de maneira simples e
- que os usuários possam utilizá-las também individualmente.

### 4. Arquitetura da primeira aplicação

Como um passo importante para a criação da Orchidea, este trabalho apresenta uma primeira proposta de aplicação. Esta aplicação proposta tem por objetivo permitir que um conjunto de pessoas, utilizando a mesma rede, possam se conectar e interagir de forma musical. Para atender as propostas apresentadas anteriormente, este trabalho partiu de uma arquitetura de sistema que pudesse atender as características consideradas necessárias para este projeto. Para atender a simplicidade de manutenção futura do código o modelo arquitetural escolhido utiliza uma arquitetura em camadas para a definição do aplicativo. Além disto, a utilização deste modelo permitiu descrever os componentes que compõem as camadas propostas mantendo um baixo acoplamento no sistema e permitindo a escolha de tecnologias específicas para a implementação de cada funcionalidade.

Com isto, o sistema foi pensado inicialmente em três camadas: Uma camada de GUI para a interação com o usuário, uma camada de aplicação para o controle das mensagens de síntese e uma camada de comunicação para a troca de mensagens entre os usuários. Esta arquitetura será apresentada a seguir.

#### 4.1. Camada de Interface Usuário/IO

Na arquitetura proposta, a Camada de Interface Usuário/IO é responsável por garantir a interação do usuário com o sistema. Para isto, esta camada mapeou os sensores do *smarthphone* e responde às entradas do usuário tanto graficamente em sua GUI quanto sonoramente por meio da emissão de sons. Por esta razão, esta camada possui três componentes: Sensores, GUI e Sintetizador.

**Sensores** é o componente responsável por mapear a interação do usuário e gerar eventos para o Gerenciador de Eventos na Camada de Aplicação. Um instrumento específico utiliza um ou mais sensores do dispositivo, o que implica em registrar-se junto a este componente para a escuta destes sensores. É importante notar que nem todo instrumento irá utilizar todos os sensores disponíveis em um aparelho e que a utilização de um instrumento específico implica em um ou mais sensores estarem disponíveis em um determinado aparelho. Também é responsabilidade deste componente definir as restrições (*constraints*) dos sensores pois diferentes dispositivos podem possuir diferentes limitações. Assim, mais do que verificar um toque na tela em determinada posição, é necessário que o instrumento faça um mapeamento deste evento em relação ao tamanho da tela de maneira a gerar um evento com significado semântico podendo afirmar, por exemplo, qual tecla do piano foi pressionada por este toque na tela [Schiaivoni 2016].

A **GUI** é o componente responsável por dar *feedback* visual ao usuário e refletir ao usuário a utilização do instrumento. Este componente recebe eventos com significado semântico do Gerenciador de Eventos de forma a refletir um evento local levando em consideração as restrições do dispositivo utilizado.

O último componente desta camada é o **Sintetizador**. Este componente é o responsável por dar *feedback* sonoro a todos os eventos que acontecem no ambiente.

#### 4.2. Camada de Aplicação

A Camada de Aplicação é responsável por gerenciar os eventos que ocorrem no aplicativo. Esta possui um componente chamado Gerenciador de Eventos, que é o responsável local pela troca de mensagens entre os demais componentes do sistema.

O **Gerenciador de Eventos** é o centro da aplicação e funciona da mesma maneira para todos os instrumentos desenvolvidos comunicando-se tanto com a Camada de Interface Usuário / IO quanto com a Camada de Comunicação. Ao receber um evento dos Sensores, o Gerenciador de Evento dispara uma mensagem de rede que sinaliza ao ambiente o evento local e dispara um evento para que a GUI reflita a interação do usuário. Desta maneira, o desenvolvimento de um Instrumento possui um baixo acoplamento entre seus Sensores e sua GUI, o que permitiu um desenvolvimento modularizado destes componentes. Ao receber uma mensagem de rede, o Gerenciador de Evento comunica esta mensagem ao Sintetizador que será responsável por refletir sonoramente um evento do ambiente.

#### 4.3. Camada de Comunicação

A Camada de Comunicação é responsável pela comunicação em rede. Para realizar esta tarefa foi definido 2 componentes, envio de mensagens e empacotador/desempacotador.

O **Empacotador/Desempacotador** é o componente de software responsável por definir as mensagens em um formato de rede de maneira a garantir a compatibilidade entre as mensagens do ambiente. Ao pensarmos na arquitetura de referência do protocolo TCP/IP, este componente definirá um protocolo de aplicação, especificando formatos de mensagens de maneira que aplicações distintas possam se comunicar por meio de um protocolo comum. Desta maneira, a definição de um instrumento também implica em definir quais mensagens o mesmo terá e como as mesmas serão empacotadas para a distribuição em rede.

O **Envio de Mensagens** é o responsável por enviar e receber as mensagens para / de os demais dispositivos conectados na rede. Tal componente, na arquitetura de referência do protocolo TCP/IP, possuirá uma implementação de um protocolo de transporte que garanta a comunicação em rede das mensagens já empacotadas.

## 5. Desenvolvimento

Uma vez definidas as funcionalidades necessárias para o desenvolvimento da aplicação por meio da definição dos componentes de cada camada, foi possível analisar um destes componentes e definir quais tecnologias seriam utilizadas para o desenvolvimento de cada parte do sistema. A plataforma Android foi escolhida inicialmente como plataforma a ser utilizada devido a sua popularidade entre os acadêmicos da Universidade Federal de São João del-Rei. A pesquisa quanto a qual plataforma de celular é a mais utilizada entre os alunos foi realizada pelo Núcleo de Tecnologia da Informação (NTInf) da Instituição.

### 5.1. Sintetizador

Foram elencadas algumas possibilidades para a geração de som na plataforma Android, como as classes nativas da linguagem Java, Sound Pool e Media Player. Como uma das propostas iniciais da ferramenta é garantir a simplicidade da evolução, consideramos a possibilidade de utilizar uma biblioteca não nativa que permitisse a síntese sonora. A *libpd* extrai a funcionalidade do ambiente de programação musical Pure Data<sup>1</sup> e o torna disponível como um *callback* de processamento de áudio em dispositivos Android. Entre as bibliotecas pesquisadas, a **libpd** [Brinkmann et al. 2011] mostrou-se bastante adequada como *engine* de síntese.

Com a utilização da *libpd* podemos desacoplar a *engine* de síntese do aplicativo e permitir que usuários deste ambiente de programação musical criem seus próprios sintetizadores a partir de um conjunto de mensagens definidas na aplicação. Com isto, a síntese e mixagem de todos os sons da orquestra de dispositivo fica sob a responsabilidade de um motor de síntese bastante maduro e eficaz.

### 5.2. GUI e Sensores

Os dispositivos Android têm um hardware limitado e uma tela de pequeno porte sendo geralmente equipados com um grande número de sensores e dispositivos de comunicação, como um microfone, wi-fi, *Bluetooth*, receptor GPS, tela *touchscreen*, sensores de

---

<sup>1</sup>Pure data é um ambiente de programação musical com o capacidade de processamento em tempo real [Puckette et al. 1996]. Mais informações podem ser encontradas em <http://puredata.inf>.

inclinação, câmera, entre outros. A fim de otimizar a gestão de todos esses recursos e lidar com as limitações do *hardware*, o sistema Android implementa um modelo de processo *multithread* em que apenas uma única *thread* pode acessar a interface de usuário, enquanto as outras funcionalidades são executadas em segundo plano [Amalfitano et al. 2011].

Nesta plataforma, o componente responsável por apresentar uma interface de usuário visual para cada tarefa é chamado Activity. As tarefas de um aplicativo Android incluem geralmente uma ou várias classes de Activity que estendem a classe base do desenvolvimento. A interface de usuário mostra cada atividade na tela e é construída usando outras classes de estrutura, tais como View, ViewGroup, Widget, Menu, diálogos, etc.

Na etapa de desenvolvimento da interface de usuário foi visto que a cada instrumento do aplicativo necessita de uma activity diferente pois mesmo que as telas não tenham muita diferença gráfica entre uma é outra, suas tarefas serão diferentes. Por esta razão, deixamos sob a responsabilidade de cada Instrumento definir sua GUI e sua comunicação com os sensores disponíveis em um dispositivo.

### 5.3. Comunicação

As decisões de projeto quanto a comunicação em rede envolveu duas questões principais: o método de endereçamento de rede a ser utilizado e a estrutura de dados para o empacotamento das mensagens de rede.

Existem alguns métodos de endereçamento para transmissão de mensagens em rede como Unicast, Broadcast e Multicast. Mensagens Unicast são trocadas entre *hosts* específicos, como um computador pessoal e um servidor Web, mensagens Broadcast são enviadas a todos os usuários de uma rede e mensagens Multicast são enviadas a apenas uma parte específica de usuários em uma rede.

Dados estas opções e o contexto deste trabalho, optamos por utilizar a comunicação em grupo (broadcast/multicast) como método de endereçamento evitando, com isto, o registro e manutenção de listas de usuários e demais detalhes necessários para a comunicação unicast.

Quanto à estrutura de dados para o empacotamento das mensagens de rede, alguns formatos poderiam ser utilizados no contexto desta aplicação como XML, JSON, OSC e texto plano. Apesar da popularidade dos formatos XML e JSON para aplicações web, o OSC (Open Sound Control) é um formato de mensagens focado no controle de aplicações musicais de tempo real [Schmeder et al. 2010]. Este foco em comunicação em tempo real evita a conversão de valores inteiros e ponto flutuantes em textos, como nos formatos XML e JSON, diminuindo com isto a largura de banda necessária para o envio de uma mensagem e aumentando a velocidade de entrega de um pacote de rede.

OSC é ainda um formato de mensagem utilizado para a comunicação em redes em diversas aplicações musicais e ambientes de programação musical, como Pure Data, Supercollider, CSound entre outros. Com isto, podemos garantir que a ferramenta seja conectável com aplicações existentes, conforme foi requisitado inicialmente.

## 6. Implementação

Uma vez definida a arquitetura do sistema e tomadas as decisões de projeto necessárias para a implementação do aplicativo, o próximo passo foi a implementação.

As camadas e componentes representados na arquitetura proposta foram implementados por um ou mais objetos neste primeiro protótipo. A camada de Interface de Usuário/IO, por exemplo, foi implementada pelos objetos Sensores e pela interface Instrumento. O objeto Sensor é o responsável por, ao iniciar a aplicação, mapear todos os sensores presentes no dispositivo. Baseado nos sensores presentes no dispositivo, o aplicativo pode apresentar ao usuário quais instrumentos podem ser utilizados neste dispositivo para que o usuário possa escolher qual instrumento deseja utilizar, como mostrado na Figura 4. Vale notar que alguns instrumentos podem não estar disponíveis ao usuário devido ao sensor utilizado pelo instrumento não estar disponível em seu dispositivo. No entanto, a síntese deste instrumento não depende de seus sensores e por isto a mesma estará disponível caso este instrumento venha a ser utilizado por outro usuário da rede.

Um dos pontos importantes de nossa implementação em relação aos requisitos propostos é a capacidade de ampliar o sistema e adicionar novos instrumentos quando necessário. Para atender a este requisito foi criada uma interface Instrumento. A interface Instrumento traz a abstração do que significa um Instrumento para o ambiente e os instrumentos desenvolvidos implementam seus métodos. Estes instrumentos são responsáveis por gerar sons baseado nos eventos recebidos do “Gerenciador de Eventos” e devem estar presentes em todas as instâncias do aplicativo. Os Instrumentos também são responsáveis por definir suas mensagens e gerenciar suas interfaces gráficas. Uma destas interfaces pode ser vista na Figura 5.

Na Interface Instrumento o método “Tocar Solo” é o responsável por permitir que o usuário toque individualmente. Já o método “tocar” é o responsável por permitir com que diversos usuários toquem utilizando uma rede local, simulando uma orquestra. Esta escolha pode ser feita no início da aplicação, como apresentado na Figura 3.

A camada de Comunicação da arquitetura proposta foi implementada na classe Rede. A classe Rede é responsável por fazer o envio das mensagens de sons para os demais dispositivos conectados na rede e é também responsável por receber as mensagens enviados por outros dispositivos. Para a primeira implementação as mensagens enviadas seguem o seguinte padrão: instrumento/nota, sendo que todos os instrumentos conectados tocam eles mesmos e os demais conectados. Assim, o instrumento local utiliza a interface de rede para enviar uma mensagem entre sua interface de usuário e seu sintetizador.

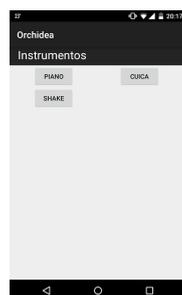
## 7. Resultados parciais

A proposta deste projeto traz uma lista de requisitos funcionais e não funcionais que a ferramenta almejada apresentar para atender às necessidades impostas. Partindo destes requisitos o próximo passo foi partir para um desenho arquitetural do aplicativo desacoplando as funcionalidades desejadas em componentes específicos.

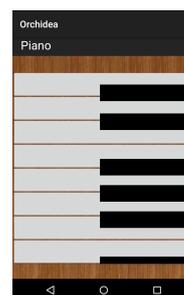
Uma vez que as funcionalidades foram desacopladas, foi possível pensar em soluções específicas para o desenvolvimento de cada uma delas. Nesta etapa foram escolhidas as seguintes tecnologias, a) a plataforma Android como o sistema operacional a qual foi desenvolvida a aplicação, escolhido por ser uma plataforma aberta e bem documen-



**Figura 3.** Possibilidade de escolha entre execução solo ou em grupo.



**Figura 4.** Seleção de instrumentos disponíveis para o dispositivo.



**Figura 5.** Piano, instrumento tocado utilizando o touchscreen.

tada para os desenvolvedores; b) libpd como a biblioteca para geração sonora e síntese, escolhida por desacoplar a síntese, geração de sons e mixagem de todos os instrumentos; c) Multicast para a transmissão de mensagens, por permitir a comunicação de rede em grupo sem aumentar consideravelmente o tráfego na rede; d) o modelo OSC para o empacotamento de mensagens por ser um formato de mensagens focado no controle de aplicações musicais de tempo real compatível com diversas ferramentas existentes para práticas musicais.

Entre as decisões de projeto apresentadas, a opção pelo endereçamento multicast, julgada inicialmente a mais adequada, se mostrou inviável pois vários dispositivos trazem uma limitação de fábrica que impede a comunicação por este modo de endereçamento. Por esta razão, tal decisão não foi seguida na implementação deste projeto sendo incluídas as possibilidades de endereçamento usando broadcast e também por meio de um servidor centralizado endereçado com unicast.

A partir do desenho arquitetural e das decisões de projetos os componentes propostos foram divididos em classes e métodos, conforme apresentado no diagrama de classe da implementação. Alguns componentes que no inicialmente foram vistos como classes foram implementados como classes ou métodos de uma classe em um refinamento da proposta arquitetural.

A implementação foi feita para duas funcionalidades diferentes o usuário poder tocar individualmente apenas com o seu próprio “*smarthphone*” e a implementação onde o usuário pode tocar com outros usuários em uma rede local, estas denominadas como Solo e Multicast da tela inicial do projeto.

## 8. Conclusão

É possível notar que o método de síntese musical, estilos e pluralidades culturais estão intrinsecamente ligados a época e a sua geração. A Geração Z, também conhecida como geração dos nativos digitais, influenciou no processo de criação e manuseio da música, incorporando os dispositivos móveis nesse processo. A possibilidade de integrar pessoas por meio de aplicações musicais para dispositivos móveis remete ao objetivo inicial destes aparelhos: a comunicação, mas amplia os limites desta comunicação permitindo e possibilitando a comunicação musical. Este artigo apresentou o projeto da ferramenta, uma aplicação para execução musical colaborativa / cooperativa em redes de computado-

res. Foi apresentado a proposta inicial da ferramenta, o modelo arquitetural proposto, a implementação desta arquitetura e as decisões de projeto e a implementação deste aplicativo<sup>2</sup>.

Findado o desenvolvimento inicial, novos desafios surgem para que a Orchidea seja uma realidade musical e não apenas tecnológica. O desenvolvimento desta aplicação deverá fortalecer uma pesquisa interdisciplinar envolvendo musicista e leigos na criação de uma orquestra de celulares na Universidade Federal de São João Del Rei como um projeto de pesquisa entre os cursos de Ciência da Computação, Música, Artes Cênicas e Artes Aplicadas. Nesse contexto, orquestra de ideias é uma metáfora para a proposta transdisciplinar de integração artística por meio da tecnologia onde a utilização de computadores e “smartphones” como instrumentos dessa orquestra, transdisciplinaridade esta que não se propõe a uma apropriação dos ritos e organizações de uma orquestra mas que pretende possibilitar a criação de música colaborativa, cotidiana [Keller 2018] que proporciona aos ouvintes uma atuação direta no processo de criação.

Diversas pessoas tem demonstrado interesse em participar da Orquestra e algumas propostas artísticas já estão em desenvolvimento. O próximo passo desta pesquisa é explorar ainda mais a portabilidade, conectividade e ubiquidade da tecnologia móvel de forma a atender prontamente os questionamentos artísticos dos participantes destes grupos. Entre as propostas estéticas e artísticas a serem implementadas estão intervenções musicais em espaços públicos, concertos que deverão contar com a participação da plateia e a utilização do corpo na dança para a criação sonora baseado nos sensores dos dispositivos móveis. Com isto, pretendemos difundir a possibilidade de criação artística musical também entre leigos e permitir que a música ubíqua se torne uma realidade no dia a dia das pessoas.

## **Trabalhos Futuros**

Apesar de a arquitetura aqui proposta propiciar um baixo acoplamento de código, ainda é necessário conhecimento em programação para adicionar novos instrumentos a esta aplicação. Pretendemos criar uma arquitetura de plugins que permita adicionar e remover instrumentos de maneira mais simples a esta aplicação de modo que não seja necessário a distribuição de uma nova versão da aplicação para a adição de novos instrumentos. É intenção também que a criação de novos instrumentos possa ser feita por leigos e não apenas por programadores experientes.

As possibilidades iniciadas com este projeto não se encerram no desenvolvimento deste aplicativo. A recente utilização desta aplicação trouxe a tona a dificuldade de disponibilizar e instalar esta aplicação nos dispositivos móveis dos participantes do projeto, especialmente para o público, o que levou a discussões sobre alternativas viáveis para a participação de membros externos nas apresentações de nosso grupo. Entre estas possibilidades encontramos a utilização de sintetizadores e meta-instrumentos criados em HTML5 e webaudio, que funcionam de maneira regular na maioria dos dispositivos testados, e que não requer instalações ou atualizações extras nos dispositivos dos participantes. Por outro lado, esta opção não permite a comunicação em rede dos participantes e a criação colaborativa, foco inicial deste projeto.

---

<sup>2</sup>Esta ferramenta encontra-se disponível no endereço  
<http://www.bitbucket.org/eduardox/poc-definitivo.git>

Uma possibilidade futura é utilizar o ambiente de programação visual Mosaiccode [Schiavoni and Gonçalves 2017], desenvolvido neste mesmo laboratório, para gerar os meta-instrumentos e coordená-los. O Mosaiccode é uma ferramenta de programação por blocos, propiciando um ambiente de computação musical livre de linguagem de programação. Este ambiente já possui capacidade para o desenvolvimento de aplicações com HTML5 e Javascript e pode ser estendido para o desenvolvimento de outras aplicações, como novos instrumentos para a Orchidea.

## Agradecimentos

Os autores gostariam de agradecer aos participantes do Grupo de Estudo em Arte Digital do Departamento de Computação da Universidade Federal de São João del-Rei, especialmente os Professores Adilson Siqueira e Rogério Tavares, e aos membros do ALICE (Arts Lab in Interfaces, Computers, and Everything Else) que auxiliaram nos debates e desenvolvimento deste projeto. O Coautor gostaria de agradecer o apoio financeiro institucional da FAPEMIG e da PROPE/UFSJ.

## Referências

- Amalfitano, D., Fasolino, A. R., and Tramontana, P. (2011). A gui crawling-based technique for android mobile application testing. In *ICST Workshops*, pages 252–261. IEEE Computer Society.
- Brinkmann, P., Inc, G., Kirn, P., Lawler, R., McCormick, C., Roth, M., and christoph Steiner, H. (2011). Embedding pure data with libpd. In *in Proceedings of the Pure Data Convention, Weimar*.
- Bukvic, I., Martin, T., Standley, E., and Matthews, M. (2010). Introducing l2ork : Linux laptop orchestra. In Beilharz, K., Bongers, B., Johnston, A., and Ferguson, S., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 170–173, Sydney, Australia.
- Dannenberg, R. B., Cavaco, S., Ang, E., Avramovic, I., Aygun, B., Baek, J., Barndollar, E., Duterte, D., Grafton, J., Hunter, R., et al. (2007). The carnegie mellon laptop orchestra. In *Proceedings of the 2007 International Computer Music Conference, vol II*, pages 340–343. The International Computer Music Association.
- Essl, G. and Rohs, M. (2007). Shamus – a sensor-based integrated mobile phone instrument. In *IN PROCEEDINGS OF THE INTERNATIONAL COMPUTER MUSIC CONFERENCE (ICMC)*, pages 27–31.
- Keefe, P. O. and Essl, G. (2011). The visual in mobile music performance. In Jensenius, A. R., Tveit, A., Godoy, R. I., and Overholt, D., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 191–196, Oslo, Norway.
- Keller, D. (2018). Challenges for a second decade of ubimus research: Knowledge transfer in ubimus activities. *Música Hodie*, 18:147–165.
- Keller, D., Lazzarini, V., and Pimenta, M. S. (2014). *Ubiquitous music*. Springer.
- Lee, S. W., Freeman, J., Colella, A., Yao, S., and Troyer, A. V. (2011). Collaborative musical improvisation in a laptop ensemble with lolc. In Goel, A. K., Harrell, D. F.,

- Magerko, B., Nagai, Y., and Prophet, J., editors, *Creativity and Cognition*, pages 361–362. ACM.
- Mathews, M. V. (1963). The digital computer as a musical instrument. *Science*, 142(3592):pp. 553–557.
- Oh, J., Herrera, J., Bryan, N. J., Dahl, L., and Wang, G. (2010). Evolving the mobile phone orchestra. In Beilharz, K., Bongers, B., Johnston, A., and Ferguson, S., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 82–87, Sydney, Australia.
- Puckette, M. et al. (1996). Pure data: another integrated computer music environment. *Proceedings of the second intercollege computer music concerts*, pages 37–41.
- Roberts, C., Forbes, A., and H'ollerer, T. (2013). Enabling multimodal mobile interfaces for musical performance. In Yeo, W., Lee, K., Sigman, A., H., J., and Wakefield, G., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 102–105, Daejeon, Republic of Korea. Graduate School of Culture Technology, KAIST.
- Schiavoni, F. L. (2016). Event-based ubiquitous music interaction with mcm: A musical communication modeling methodology. In *International Symposium on Computer Music Multidisciplinary Research*, pages 284–298. Springer.
- Schiavoni, F. L. and Gonçalves, L. L. (2017). Programação musical para a web com o mosaicode. In *Anais do XXVII Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, pages 1–6, Campinas - SP - Brazil.
- Schiemer, G. and Havryliv, M. (2005). Pocket gamelan: A pure data interface for mobile phones. In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, NIME '05, pages 156–159, Singapore, Singapore. National University of Singapore.
- Schmeder, A., Freed, A., and Wessel, D. (2010). Best practices for open sound control. In *Linux Audio Conference*, Utrecht, NL.
- Tanaka, A. (2004). Mobile music making. In Nagashima, Y., Ito, Y., and Furuta, Y., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 154–156, Hamamatsu, Japan.
- Trueman, D. (2007). Why a laptop orchestra? *Organised Sound*, 12(2):171–179.
- Tsabary, E. (2014). Music education through innovation: The concordia laptop orchestra as a model for transformational education. *INTED2014 Proceedings*, pages 657–664.
- Wang, G., Bryan, N., Oh, J., and Hamilton, R. (2009). Stanford laptop orchestra (slork). In *International Computer Music Conference*, Montreal.
- Wang, G., Essl, G., Telekom, D., and Penttinen, H. (2008). Mopho: Do mobile phones dream of electric orchestras. In *Proc. Intl. Computer Music Conference (ICMC 2008)*, page 29.