

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUCAS FIALHO ZAWACKI

**Um Sistema Online para Gravação de
Sintetizadores Analógicos**

Trabalho Final Apresentado em Cumprimento
Parcial dos Requerimentos para Obtenção de Grau
de Bacharel em Ciência da Computação

Prof. Dr. Marcelo Oliveira Johann
Orientador

Porto Alegre, dezembro de 2012

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Prof^a. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	5
LISTA DE FIGURAS	6
LISTA DE TABELAS	7
RESUMO	8
ABSTRACT	9
AGRADECIMENTOS	11
1 INTRODUÇÃO	12
1.1 Motivação	12
1.1.1 Produção Musical Ubíqua	13
1.2 Sintetizadores como Servidores de Gravação	14
1.3 Organização deste trabalho	14
2 SINTETIZADORES	15
2.1 Métodos de síntese	15
2.2 Uma breve história dos sintetizadores eletrônicos	15
2.3 Funcionamento Básico de um Sintetizador	18
3 GRAVAÇÃO REMOTA DE ÁUDIO	19
3.1 Estado da Arte em Performance por Rede	19
3.2 Transmissão de Áudio: Tempo real e em lotes	19
3.3 Tempo real e problemas com acesso	20
3.4 Solução com processamento em lotes	20
3.5 Problemas com acesso remoto	21
3.6 Vantagens do acesso remoto em lotes	22
4 NOTAÇÃO MUSICAL ELETRÔNICA	23
4.1 O Protocolo MIDI	23
4.2 Arquivos MIDI, Standard MIDI files	24
5 ANALOG FARM	26
5.1 Uso típico do sistema	26
5.2 O Serviço	27
5.3 Interface WEB	28
5.4 Interface Webservice REST	28

6	DETALHES DE IMPLEMENTAÇÃO	31
6.1	Banco de Dados	31
6.2	Tarefas	31
6.3	Gravações	33
6.3.1	Obtenção de gravações	33
6.3.2	Hardware de gravação	34
6.4	Distribuição de arquivos	35
6.5	Automatização de Implantação, Testes e Monitoramento	35
6.5.1	Implantação do Software	36
6.5.2	Monitoramento	37
6.5.3	Painel de Controle	38
6.5.4	Testes Unitários, de Integração e de Validação	38
6.5.5	Testes de Carga	39
6.5.6	Testes da Interface REST	39
6.6	Desafios técnicos e Soluções	40
6.6.1	Diferenças entre um servidor tradicional e um servidor de gravações	40
6.6.2	Servidor de gravação distribuído	40
7	CONCLUSÃO	41
7.1	Trabalhos futuros	41
7.1.1	Instrumentos musicais e equipamentos como servidores	41
7.1.2	Melhorias no Analog Farm	42
	REFERÊNCIAS	43

LISTA DE ABREVIATURAS E SIGLAS

MIDI	Musical Instrument Digital Interface
SMF	Standard MIDI File
DAW	Digital Audio Workstation
FLAC	Free Lossless Audio Codec
REST	Representational State Transfer
API	Abstract Programming Interface
DSP	Digital Signal Processing
VST	Virtual Studio Technology
URL	Universal Resource Locator

LISTA DE FIGURAS

Figura 2.1:	Diagrama de funcionamento de um sintetizador	18
Figura 5.1:	Diagrama da Arquitetura do Analog Farm	27
Figura 5.2:	Tela de criação de uma nova gravação	28
Figura 5.3:	Tela de listagem de músicas e gravações	29
Figura 5.4:	Tela de obtenção de gravação	29
Figura 6.1:	Arquitetura de Software do Analog Farm	31
Figura 6.2:	Esquema do Banco de Dados do Analog Farm	32
Figura 6.3:	Servidor onde está hospedado o Analog Farm	34
Figura 6.4:	Interface de gravação, Roland MKS-50, Fonte de Alimentação e Studio Electronics ATC-1	35

LISTA DE TABELAS

Tabela 1.1:	Preços de sintetizadores analógicos em Maio de 2012	13
Tabela 2.1:	Diferentes Métodos de Síntese Sonora	16
Tabela 4.1:	Formatos de notação e áudio suportados por DAWs comuns	24
Tabela 4.2:	Comparação do tamanho de diferentes formatos de música	25
Tabela 5.1:	Especificação do Webservice REST do Analog Farm	30

RESUMO

Este trabalho propõe um sistema composto de *hardware* e *software* que permite o acesso remoto a sintetizadores analógicos usando a rede mundial de computadores. Neste sistema, um arquivo *MIDI* com a composição do usuário é enviado a um servidor *web*, é tocado em um sintetizador analógico e uma gravação de alta qualidade é disponibilizada. A arquitetura proposta implementa acesso às máquinas por meio de processamento em lotes com o intuito de maximizar a utilização do *hardware* e em decorrência permitir um menor custo de acesso aos usuários finais. O principal objetivo é tornar máquinas raras e caras disponíveis para gravações de alta qualidade ao menor custo possível.

Palavras-chave: Sintetizadores analógicos, computação e música, MIDI, gravação.

An Online System For Recording Analog Synthesizers

ABSTRACT

This paper proposes a system composed by *hardware* and *software* which enables remote access to analog synthesizers by means of the world wide web. This system uses *MIDI* files representing the user compositions which are sent to a server, played back by a real analog synthesizer and a high quality recording is then made available. The proposed architecture implements batched access to the machines with the intention of maximizing the usage and by result allowing a smaller cost of access to end users. The goal of this work is to make expensive and rare machines available for high quality recording at lowest possible cost.

Keywords: analog synthesizers, recording, MIDI, computer music.

“Será que o sintetizador sintetiza som quando não tem ninguém para ouvi-lo?”

AGRADECIMENTOS

Aos meus professores e colegas que dividiram todos esses anos na faculdade comigo e me ajudaram a concretizar este trabalho. À minha família por me apoiar e prover comida e abrigo. À minha querida namorada pelo carinho, apoio e por relevar todas as noites de sábado que eu passei estudando e escrevendo este texto.

1 INTRODUÇÃO

Os primeiros sintetizadores analógicos, aparelhos que usam componentes eletrônicos como geradores de função e filtros para criar sinais sonoros, foram muito usados na música das décadas de 70 e 80. Porém com a popularização dos sintetizadores digitais e do uso de computadores para a síntese sonora os aparelhos analógicos foram considerados por um certo período apenas como desatualizados. Entretanto, ultimamente, o interesse em sintetizadores analógicos vem crescendo novamente e o seu uso é cada vez mais comum na música moderna, por apresentarem diferenças muito significativas nas características sonoras em relação aos substitutos digitais. Infelizmente, se por um lado existe este interesse renovado, por outro lado as peculiaridades associadas à produção destas máquinas e o nicho ainda limitado do mercado torna-as bastante caras, caras o suficiente para impedir que estúdios de pequeno/médio porte ou músicos sem muitos recursos financeiros tenham acesso a esta tecnologia. A proposta deste trabalho é a criação de um sistema online que permita a usuários o acesso a sintetizadores analógicos operando como servidores. O sistema funciona recebendo arquivos *MIDI* de notação musical que serão processados em lotes pelo servidor, tocados pelos sintetizadores, gravados e disponibilizados como uma gravação de alta qualidade para download.

Ao longo do trabalho serão discutidas as motivações e o contexto deste sistema, serão apresentadas as escolhas de projeto e uma descrição detalhada da implementação. Ao fim serão apresentadas algumas perspectivas, trabalhos futuros e possibilidades abertas ao encarar instrumentos musicais como servidores de gravações.

1.1 Motivação

O consumo e a produção de música são aspectos muito presentes no nosso cotidiano e, devido aos meios de distribuição e produção digital de conteúdo, a identidade de consumidor e produtor fica cada vez mais indistinguível. Em 2010 um estudo fez uma amostragem de 2.5 milhões de vídeos do Youtube (SYSOMOS, 2010) e descobriu que a categoria mais proeminente entre eles era *música*, com surpreendentes 31% do total. Segundo o próprio Youtube (Google Inc, 2012) o *site* recebe o *upload* de aproximadamente 1 hora de vídeo por minuto o que demonstra a quantidade de conteúdo novo produzida também é enorme e como visto anteriormente, grande parte deste é música. Na mesma linha temos o popularidade de *sites* como o *Soundcloud*, uma rede social voltada para a produção musical, a qual em 2012 ultrapassou a marca de 10 milhões de usuários (SOUNDCLOUD, 2012). De fato em (PERPETUO; SILVEIRA, 2009) observa-se que no contexto dessas redes, a divisão entre fã e artista se torna obsoleta.

Estas novas tendências de produção alimentam um mercado de *Digital Audio Workstations (DAWs)* e instrumentos virtuais comercializados para músicos, tanto amadores

Modelo	Fabricante/Marca	Preço (U\$)	Polifonia	Em produção?
Voyager Rck	Moog Music	\$2,795.00	monofônico	sim
SE-1X	Studio Electronics	\$1,679.00	monofônico	sim
Omega8	Studio Electronics	\$4,999.00	polifônico	sim
Prophet Five	Seq. Circuits	\$4,258.00	polifônico	usado
CS80	Yamaha	\$18,249.00	polifônico	usado
Jupiter8	Roland	\$2,699.00	polifônico	usado
Odyssey	ARP	\$2,529.00	monofônico	usado

Tabela 1.1: Preços de sintetizadores analógicos em Maio de 2012

quanto profissionais. Logo, técnicas de composição e gravação eletrônicas hoje têm uma grande representatividade nessa nova economia baseada na criação de conteúdo. Estas tecnologias resolvem um problema de *acesso* a ferramentas de produção e *distribuição* de conteúdo e demonstram que qualidade não está necessariamente relacionada com altos custos e processos profissionais.

Entretanto o filão que informalmente chamamos de “indústria musical” no que tange a produção profissional ainda emprega principalmente tecnologias mais antigas do que a revolução digital como: instrumentos acústicos, mesas de mixagem, salas de gravação, sintetizadores analógicos, entre outros. Como já observado, a produção de gravações com estes equipamentos não é mais requisito para alta qualidade, mas acredita-se que se o músico deseja usá-los isto deveria ser possível. Porém o acesso a estes é limitado para músicos amadores, principalmente por razões econômicas. Como exemplo, na tabela 1.1 estão listados alguns preços pesquisados dia 20 de Maio de 2012 para sintetizadores icônicos e utilizados ainda hoje. Dado a variedade de máquinas diferentes e alto preço de aquisição de cada uma, nota-se porque ainda existe um problema de acesso deste recurso.

A *motivação* para este trabalho então é a possibilidade de utilizar a tecnologia e o poder computacional existentes hoje para diminuir este abismo entre as diferentes técnicas de produção e aproximar todo o tipo de compositor destes instrumentos inacessíveis, seja por causa de sua raridade, localização geográfica ou preço elevado. Objetiva-se juntar o caráter diferenciado dos instrumentos analógicos e antigos da praticidade e baixo custo dos *plugins* e instrumentos virtuais.

1.1.1 Produção Musical Ubíqua

Em (PIMENTA et al., 2009) são apresentadas as ideias de computação e música imersos em aspectos cotidianos e fazendo uso de celulares e *gadgets* que hoje possuem capacidades computacionais suficientes para trabalho com áudio. Acredita-se que este trabalho está resolvendo parte do problema de acesso às ferramentas que um músico pode querer usar para produzir peças com qualidade. Este problema está sendo resolvido de maneira a expor para o músico uma interface *web* que pode ser facilmente acessada a partir de uma grande quantidade de dispositivos e assim fazendo da produção musical também um processo mais ubíquo (ZAWACKI; JOHANN, 2012a).

Além disto a existência de servidores de instrumentos analógicos ajuda a promover o uso desta classe diferenciada de instrumentos, aumenta o conhecimento público sobre eles e alimenta o mercado de instrumentos virtuais e emulações em *software* destas características sonoras.

1.2 Sintetizadores como Servidores de Gravação

Temos então um crescente mercado de músicos amadores e profissionais que produzem grandes volumes de música, em diversos tipos de dispositivos. Ao mesmo tempo a busca por qualidade das gravações e diferentes sonoridades é evidente no mercado, dado a grande disponibilidade de *softwares* de gravação, instrumentos virtuais e *plugins*. Nessa conjuntura se configura o problema do acesso dos diferentes instrumentos de produção musical, no caso deste trabalho os sintetizadores analógicos, por todo tipo de músicos, principalmente os amadores.

Este trabalho objetiva resolver o problema do acesso aos sintetizadores analógicos e diminuir a distância que existe entre os músicos amadores e profissionais. Para isto será apresentado um sistema de acesso *online* a sintetizadores analógicos, usando uma abordagem não em tempo real para a gravação de músicas e uma interface *web* e *webservice*. As características deste sistema devem potencializar uma máxima utilização dos instrumentos sendo gravados e uma melhor integração com o processo de composição usado pelos usuários.

O resultado do projeto de tal sistema é o *Analog Farm*, um serviço que permite o acesso a dois sintetizadores analógicos por meio de uma interface *web*. O sistema já se encontra em produção com um pequeno número de usuários e demonstra a viabilidade do modelo apresentado neste trabalho.

1.3 Organização deste trabalho

No segundo capítulo será apresentada uma introdução aos sintetizadores no intuito de demonstrar a sua importância na produção de música moderna. No terceiro capítulo segue-se uma rápida discussão das tecnologias de transmissão de áudio pela *internet*, suas limitações e a motivação por trás da escolha do sistema de processamento em lotes. No quarto capítulo será discutido o controle de instrumentos eletrônicos e formatos de notação musical, dois conceitos chave na implementação deste sistema.

Os capítulos cinco e seis trazem à discussão detalhes de projeto, implementação e uso do *Analog Farm*, um serviço *online* exposto como uma interface *web* e um *webservice* que permite o acesso à gravação com dois sintetizadores analógicos. No capítulo número sete são apresentadas perspectivas futuras de trabalho e outras aplicações mais abrangentes para tal sistema.

2 SINTETIZADORES

Neste contexto, sintetizadores sonoros são máquinas que geram sons eletronicamente e são frequentemente usados em peças de música moderna. Este trabalho trata principalmente de sintetizadores analógicos, portanto dispositivos de *hardware*. Também são considerados como sintetizadores sonoros os *softwares* executados em computadores digitais que criam sons com processamento de sinais (*DSP*). Neste trabalho, estes serão referidos como “instrumentos virtuais”, uma nomenclatura adotada informalmente por compositores de música eletrônica (STEINBERG, 1999).

O resto da seção inicial será usado para introduzir a ideia básica por trás da construção e uso de sintetizadores, um breve histórico desta classe de instrumentos, bem como sua importância cultural e uso na produção musical.

2.1 Métodos de síntese

A síntese de sinais sonoros pode ser feita de diversas formas. Os métodos mais fundamentais e importantes são : síntese aditiva (REID, 2000a), síntese subtrativa (Moog Music Inc, 2012a), síntese por modulação de frequência (REID, 2000b), síntese *wavetable* (BRISTOW-JOHNSON, 1996) e síntese por reprodução de amostras sonoras (COOK, 2002).

Na tabela 2.1 podem ser vistos os métodos mais comuns de síntese acompanhados dos sintetizadores mais famosos a empregar o seu uso. Sintetizadores analógicos geralmente estão associados com síntese aditiva e subtrativa visto que têm a capacidade de lidar com o som apenas como ondas elétricas, sem se envolver com conversões analógico/digitais. Além disto, as ondas e filtros presentes em sintetizadores analógicos soam diferentemente dos mesmos implementados usando modelos simplificados com processamento digital de sinais. Diversos parâmetros físicos dos componentes usados também influenciam no formato das ondas, na resposta dos filtros e outros fatores da sonoridade. Um grande esforço tem sido feito para desenvolver modelos que capturem as características agradáveis de sintetizadores analógicos (ROSSUM, 1992), com significativos avanços. Ainda assim, o som produzido por sintetizadores analógicos é visto de maneira diferenciada por músicos (PINCH; REINECKE, 2009) e seu uso é priorizado e deve continuar sendo por um longo período de tempo.

2.2 Uma breve história dos sintetizadores eletrônicos

Os primeiros sintetizadores sonoros datam do início do século 20 com a invenção do *Tellharmonium* em 1897 (WEIDENAAR, 1984), um instrumento musical eletromecânico

Síntese	Método	Sintetizadores/Ano de lançamento
Aditiva	Soma de diversos componentes parciais de ondas, geralmente harmonicamente relacionados, para a criação de uma onda de espectro mais complexo	Tellharmonium (1897), Hammond B-3 (1934)
Subtrativa	Começando de uma onda com espectro rico em frequências, aplica-se diferentes filtros para criar espectros variados de onda	Minimoog (1972), ARP 2600 (1971)
Modulação de Frequência	Uso de uma onda principal que tem sua frequência variada de acordo com frequência de alguma outra onda	Yamaha DX7 (1983)
Reprodução de Amostras	Reprodução de sons de instrumentos pré-gravados com modificações como mudança de frequência, volume e aplicação de filtros em tempo real	Korg-M1 (1988)
Wavetable	Método baseado na repetição periódica de um formato de onda pré-armazenado	Waldorf WAVE (1993)

Tabela 2.1: Diferentes Métodos de Síntese Sonora

baseado em síntese aditiva que gerava ondas elétricas usando *tonewheels*, aparatos baseado em motores AC e captadores. As ondas geradas eram transmitidas pela linha telefônica, um prenúncio interessante de transmissão moderna de áudio, e podiam ser ouvidas com um tipo primitivo de alto-falante. O Tellharmonium não foi um sucesso comercial, mas em 1934 o órgão Hammond empregou os mesmos princípios para criar um instrumento similar aos órgãos de tubo, com o intuito de ser vendido a igrejas (BARRY, 1974). Este tipo de órgão foi também adotado por músicos de blues, jazz e rock progressivo e o modelo *Hammond B-3* é um dos sintetizadores com o som mais reconhecido na música popular, principalmente quando tocado juntamente com a caixa *Leslie*, um tipo especial de alto-falante com sonofletores rotatórios.

Após a invenção dos primeiros amplificadores de válvula eletrônica diversos novos instrumentos eletrônicos foram criados. O mais famoso destes é com certeza o *Theremin* (MARTIN et al., 2001), o qual era revolucionariamente controlado com o movimento dos braços do músico relativos a duas antenas de metal. A partir de 1930, diversos componentes como filtros, osciladores e envelopes de onda já estavam sendo utilizados em instrumentos musicais eletrônicos e formariam a base do método de síntese conhecido como síntese subtrativa. Durante as décadas de 1950 e 1960 surgiram os primeiros sintetizadores modulares (BODE, 1961a), máquinas consistindo de vários componentes - dispostos em módulos separados - e conectados entre si formando um fluxo de controle, geração e processamento de ondas elétricas. Nesta época, a evolução da eletrônica (BODE, 1961b) permitiu que os primeiros sintetizadores portáteis fossem comercializados ao público por empresas como a *Moog Music* e *ARP*. Durante este período diversos artistas utilizaram estas novas tecnologias em seus trabalhos, sendo um dos mais notáveis o álbum *Switched On Bach* (CARLOS, 1968) por Wendy Carlos, que apresenta reinterpretações de composições de *Bach* usando sintetizadores *Moog*.

Na década de 70 John Chowning desenvolveu um método de síntese por modulação de frequência, o qual foi licenciado pela Yamaha em meados de 1975 e deu origem ao sintetizador DX7 (Yamaha Corporation, 1983) que marcou a música popular durante os anos 80. A síntese por modulação de frequência (popularmente chamada de síntese *FM*) também teve um papel importante na geração sonora das primeiras placas de som para computadores pessoais. Em 1983 começam a ser introduzidos os primeiros instrumentos eletrônicos com suporte a MIDI como uma maneira padrão de controle e automatização.

A partir de 1990 com o aumento da eficiência computacional em computadores digitais e a popularização dos *PCs* começam a surgir sintetizadores implementados em software e se estabelece o padrão VST (STEINBERG, 1999) de processamento de sinais e instrumentos virtuais. Outros métodos de síntese como a baseada em amostras de áudio pré-gravada também se tornaram viáveis e ganharam bastante espaço. Estes avanços da tecnologia digital somados ao alto custo de produção dos antigos sintetizadores analógicos tornaram estas máquinas mais caras e escassas e levaram ao contexto atual onde eles representam um nicho limitado do mercado. Entretanto hoje existe um interesse renovado neste tipo de dispositivo pelo reconhecimento da sua superioridade sonora para diversos tipos de sonoridades. Índícios desse interesse são os instrumentos virtuais baseados em emulações de equipamento analógico clássico e sintetizadores analógicos modernos como o *Moog Voyager* (Moog Music Inc, 2012b), *Arturia Minibrute* (REID, 2012), a linha *Korg Monotron* (KORG, 2012) entre outros.

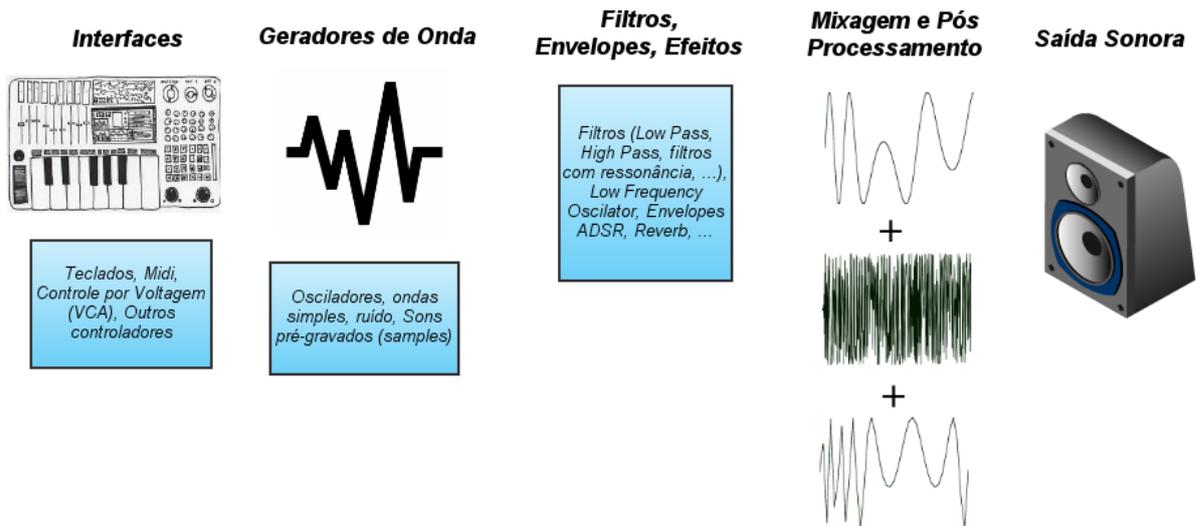


Figura 2.1: Diagrama de funcionamento de um sintetizador

2.3 Funcionamento Básico de um Sintetizador

Sintetizadores sonoros são compostos de diversos elementos, sendo eles comumente: interface de controle, geradores de áudio, processamento sonoro e saída sonora. Estes dispositivos tipicamente usam osciladores para criar uma ou mais ondas como base sonora e então aplicam nelas filtros, modulações e envelopes de onda. Posteriormente são aplicados efeitos como reverberação, compressão de dinâmica, eco e etc... Por fim a onda resultante é repassada para alto-falantes onde pode ser ouvida ou para algum equipamento de gravação como, por exemplo, uma fita ou um conversor analógico digital. Um diagrama mais completo do processo pode ser visto na figura 2.1.

Classicamente sintetizadores são controlados usando um teclado similar ao de um piano (MOOG; RHEA, 1990), que quando pressionado ajusta a frequência dos seus osciladores, criando sons de uma maneira consistente com a abstração de **nota musical** da teoria musical ocidental (em inglês *pitch*) (KLAPURI; DAVY, 2006). Frequentemente seus efeitos, filtros e demais componentes podem ter os parâmetros editados e controlados por meio de botões, potenciômetros e deslizadores. Adicionalmente, é comum que sintetizadores incluam alavancas de *pitch bending* e modulação que permitem um controle mais tátil destes valores.

Desta forma sintetizadores podem ser considerados instrumentos musicais capazes de grande expressividade (MOOG; RHEA, 1990). Isto e o fato de serem capazes de gerar sons dificilmente encontrados na natureza e em outros instrumentos acústicos faz deles uma escolha popular em gravações e performances ao vivo.

3 GRAVAÇÃO REMOTA DE ÁUDIO

Este capítulo analisa um pouco do que o estado da arte em *streaming* de música e performance por rede nos oferecem, as escolhas de projeto que precisam ser feitas para sua implementação e os motivos porque acredita-se que uma abordagem com processamento em lotes é mais propícia para o processo de gravação de sintetizadores remotamente.

3.1 Estado da Arte em Performance por Rede

Transmissão de voz pela rede e em tempo real é um problema com soluções abundantes e gratuitamente disponíveis, como por exemplo o Skype (BASET; SCHULZRINNE, 2004) e o MConf (ROESLER et al., 2012), o que dá a entender que ele já esteja resolvido com uma performance satisfatória, pelo menos em condições normais de qualidade de rede. Infelizmente o problema da transmissão de música, geralmente conhecido pelo termo *networked performance* (RENAUD; CARÔT; REBELO, 2007), eleva um pouco mais o limite do que pode ser chamado de satisfatório. Por exemplo, enquanto os protocolos de transmissão de voz se preocupam apenas em transferir fielmente frequências na faixa ocupada pela voz humana, geralmente espera-se de uma transmissão de áudio a boa qualidade e presença inteira do espectro audível.

A latência de rede para localidades dispersas geograficamente geralmente não é suficiente para sincronizar uma performance transferida pela rede (BARBOSA, 2003) e para que um músico controle um instrumento remoto, receba a resposta e seu cérebro perceba isto como “instantâneo”. Existem sistemas bastante sofisticados e caros que atacam o problema (POLYCOM, 2011) assumindo que os pontos conectados remotamente tenham uma conexão proprietária e com grande largura de banda. Existem ainda outras soluções que funcionam bem para computadores em redes locais (CÁCERES; CHAFE, 2010) e ainda outras voltadas para redes *ADSL*, mesmo que comprometendo um pouco da qualidade e/ou impondo algumas regras sobre os usuários. Um exemplo disto é *O Networkshop* (JORDÁ; BARBOSA, 2001) que descreve uma situação onde músicos colaboram entre si para criar uma performance ao vivo, estando remotamente localizados e lidando com latências na ordem de 100ms. Isto foi realizado usando um instrumento baseado em osciladores de baixa frequência que permitiram um grande tempo de reação e não deixaram transparecer variações no ritmo ou no controle.

3.2 Transmissão de Áudio: Tempo real e em lotes

Visa-se com o sistema aqui proposto o controle de sintetizadores analógicos localizados remotamente e com acesso em não tempo real, de tal forma que dado um público

suficientemente grande, eles estejam sendo utilizados para sintetizar sons durante boa parte do tempo e os usuários tenham acesso a suas gravações em um tempo futuro não muito grande. Esta é uma diferença chave desta proposta para projetos como (MAYTON; JOLIAT; PARADISO, 2012) que permite o acesso de diversos usuários simultaneamente ao sintetizador tanto como ouvintes quanto como atuadores. Com esta proposta e a implementação de um serviço chamado *Analog Farm* o que se deseja é permitir a cada músico controlar totalmente o sintetizador, **apenas durante o tempo que lhe for necessário** para a execução de sua música e receber uma cópia de alta qualidade da gravação realizada em decorrência deste controle. Portanto nossos públicos, preocupações e decisões de projeto são diferentes. Por exemplo não existe a priori a necessidade de transmissão em tempo real do som gerado, mas existe a preocupação com a qualidade final da gravação e a facilidade de aquisição desta pelo usuário. Ao mesmo tempo, considerando que o usuário não estará ouvindo o resultado “ao vivo”, é inútil fornecer o controle do sintetizador em tempo real. Ao invés disto nesse sistema opta-se por aceitar arquivos descrevendo a performance da música, algo muito comum desde a invenção dos primeiros sintetizadores e sequenciadores.

3.3 Tempo real e problemas com acesso

Caso desejássemos implementar um sistema de controle de sintetizadores analógicos em tempo real e com *streaming* dos resultados, o primeiro problema não seria necessariamente tecnológico. Em primeiro lugar seria preciso implementar um controle do acesso aos sintetizadores e dar a cada músico um espaço de tempo em que ele deve controlar o sintetizador a fim de obter sua gravação. Logo seria estabelecida uma agenda com espaços de tempo limitados para cada músico utilizar o equipamento, configurá-lo de acordo com suas necessidades e fazer a performance de sua música.

O problema inerente deste sistema de “aluguel” do instrumento, para fazer uma analogia com o que acontece com o aluguel de estúdios para a gravação sonora, é que tanto aquele que provê o sintetizador quanto aquele que o usa perdem tempo desnecessariamente. Aquele que deseja acesso ao instrumento precisa adaptar o seu horário à agenda e mesmo que seu objetivo seja apenas gravar 5 minutos de resultados sonoros ele provavelmente irá reservar um tempo muito maior para compensar por tempo de configuração, erros na performance e outros atrasos e imprevistos que podem acontecer. Já o locador do instrumento deve fazer uma troca entre dificuldade de alocação do horário na agenda, ou o que é mais comum, locar grandes espaços de tempo para cada usuário e diminuir tanto o tempo em que os instrumentos estão sendo usados para realmente gerar as gravações quanto o número de diferentes usuários do sistema.

Considerando que alguns sintetizadores analógicos - principalmente aqueles que são mais procurados - podem ser um investimento caro, este desperdício de tempo acarreta num aumento do preço do serviço. Em suma, quanto mais valor damos ao tempo em que nosso instrumento está produzindo sons, mais valor estamos perdendo ao fornecer uma infra-estrutura de acesso em tempo real a ele. Isto leva-nos a discussão do próximo tópico, acesso em lotes ao instrumento “alugado”.

3.4 Solução com processamento em lotes

É uma prática comum para os músicos armazenarem suas composições usando o formato MIDI, tenham elas sido escritas à mão usando algum tipo de *software* de notação

musical, ou sido gravadas usando alguma interface de controle e um *software* de captura de mensagens *MIDI*. Tendo esta versão da composição é possível que alguns músicos escolham por executá-la quando tiverem acesso ao sintetizador e portanto economizem algum tempo que seria gasto com preparação e performance da peça. Indo um pouco mais além, o que se propõe é que os músicos usem seus arquivos de notação e uma interface *web* para configurar o sintetizador e algumas questões sonoras das suas músicas e requisitem que a gravação seja “agendada” no sistema, para que gravação seja realizada tão logo o sintetizador esteja disponível.

Cada uma das músicas que precisam ser gravadas fica esperando numa fila e os usuários são notificados e recebem um endereço para *download* quando ela está pronta, sem a necessidade de acompanharem parte alguma do processo, visto que ele acontece automaticamente no servidor. Idealmente o tempo gasto por cada gravação é exatamente o tempo da música que precisa ser gravada e isto permite potencialmente muito mais usuários sejam atendidos e o sintetizador esteja sempre gerando valor. Na implementação do *Analog Farm* o tempo de processamento de cada tarefa é igual ao tempo da música mais aproximadamente 20 segundos para tarefas de configuração e manutenção do sintetizador chegando muito próximo do resultado desejado.

3.5 Problemas com acesso remoto

Uma das maiores diferenças de ter acesso a um sintetizador por meio de um sistema de tarefas em lotes ao invés de sentar a frente dele é a previsão do som. Interação em tempo real e reação ao som que está sendo gerado não são possíveis. Outro problema é a programabilidade dos parâmetros dos sintetizadores. Mesmo que *MIDI* seja um padrão capaz de comunicar as mudanças destes parâmetros, a maneira como estas mudanças são implementadas varia de uma máquina para a outra e, por exemplo, a modulação de algum parâmetro em um sintetizador pode ser mais acentuada do que em outro. Para um compositor pode ser importante ter um controle mais acurado e preciso dos aspectos programáveis de cada sintetizador e isto é algo que precisa ser levado em conta para que o sistema seja útil. Existem algumas opções possíveis para contornar este problema:

- Prover sons pré-gravados ao usuário: Oferecer pelo menos uma amostra para cada timbre diferente e para diferentes tipos de articulações e modulações do som. Desta forma o usuário pode se preparar ouvindo o som do sintetizador e aprendendo antes da verdadeira gravação como ele soa.
- Prévia de uma pequena parte da música antes da gravação: Permitir ao usuário ouvir alguns segundos da gravação antes de realizá-la de forma completa. Desta forma ele pode experimentar com diferentes parâmetros e se acostumar com coisas como volume, oitava das notas e etc.
- Instrumentos virtuais e emuladores em *software*: Pode ser disponibilizada uma imitação em *software* do sintetizador em questão, de forma que esta seja pelo menos boa o bastante para responder às mudanças dos parâmetros e refleti-las no som sintetizado, de tal sorte que, o emulador possa ser usado para experimentação e como referência. Desta forma o usuário é capaz de preparar sua música ouvindo esta cópia do sintetizador original, mas pode obter o som de verdade ao submeter sua composição para o processo de gravação.

Na implementação do *Analog Farm* a solução usada, no momento de escrita deste texto, é a disponibilização de diversos sons pré-gravados para cada sintetizador. Além disso, durante o período de testes percebe-se que um problema comum causado pela falta de interatividade é a gravação da música com a “oitava base” errada, isto quer dizer, ao usar um determinado timbre a música soa aguda ou grave demais, sendo o segundo muito mais recorrente por causa dos timbres graves presentes nos sintetizadores analógicos. Uma pequena prévia da música poderia resolver o problema, mas a solução utilizada é disponibilizar uma amostra da primeira nota da música usando o sintetizador e timbre selecionado, bem como as opções de transpor a música acima uma oitava (deixando-a mais aguda) ou abaixo uma oitava (deixando-a mais grave).

3.6 Vantagens do acesso remoto em lotes

Além da utilização máxima e decorrente possibilidade de servir mais usuários, existem algumas outras vantagens únicas oferecidas pelo acesso remoto a sintetizadores neste modelo em lotes. Uma delas é a centralização da estrutura de gravação e a possibilidade de investimento mais pesado nela, como visto em (ZAWACKI; JOHANN, 2012b). Grande parte dos equipamentos que tornam a montagem de um estúdio cara, como mesas de mixagem e conversores analógico/digital especializados (*analog to digital converter* ou *ADC*) não precisam ser replicados 100 vezes para atender 100 usuários, pois estes usuários estarão recebendo gravações de um servidor central onde estes equipamentos estarão instalados, num paralelo interessante com o que se considera hoje computação em nuvem. Este simples fato torna possível investimentos pesados em inovações na área de gravações e de novos instrumentos musicais, pois o resultado agora tem a capacidade de atingir um público muito maior. Indo um pouco mais além, também em (ZAWACKI; JOHANN, 2012a) é considerada a possibilidade de transformar instrumentos musicais únicos em servidores, como por exemplo o órgão de tubo na cidade de Melbourne nos Estados Unidos (City of Melbourne, 2012) que teve a instalação de uma interface de controle *MIDI* subsidiada pela prefeitura para permitir usos por artistas. Outra possibilidade interessante é a disponibilização de orquestras inteiras de instrumentos robóticos como em (MAES; RAES; ROGERS, 2011) para acesso remoto.

4 NOTAÇÃO MUSICAL ELETRÔNICA

Desde a invenção dos primeiros sintetizadores modulares criou-se um interesse em separar a interface de controle dos módulos geradores de som, primeiramente usando a tecnologia de controle por voltagem (*CV/Gate*), passando por vários formatos proprietários (GOOD, 2001) e mais tarde culminando no padrão industrial *MIDI*. Parte central deste trabalho está em prover um método de controle que seja ao mesmo tempo capaz de extrair o potencial sonoro do sintetizador e fácil de ser usado por pessoas interessadas em compor para estas máquinas. Em outras palavras o formato deve maximizar a *utilidade* do sistema por um usuário interessado sem prejudicar a sua *usabilidade*.

Na tabela 4.1 estão apresentados 10 *DAWs* comumente usados com sintetizadores e listados os formatos de arquivos usados por eles para notação musical. Estes softwares foram escolhidos baseados numa pesquisa realizada com 10.152 visitantes do *site* Synthtopia (SYNTHTOPIA, 2012), especializado em artigos sobre sintetizadores. As informações sobre formatos foram retiradas dos respectivos manuais de cada *DAW*.

O que se conclui é que todos esses ambientes de gravação apresentam suporte a arquivos *MIDI* e isto não é uma surpresa, visto que na questão do controle de sintetizadores e instrumentos virtuais ele é um padrão muito bem estabelecido e praticamente todo *software* e *hardware* lançados após sua criação são compatíveis com ele. Este suporte massivo dos *DAWs*, as ferramentas que os músicos usam em seus computadores para fazer as suas gravações juntamente com o suporte por parte dos sintetizadores faz do *MIDI* uma escolha natural para uso no Analog Farm.

A sessão a seguir contém uma rápida introdução ao protocolo *MIDI* e como ele é usado para armazenar composições em notação eletrônica, visto que ele desempenha um papel vital neste trabalho. Na sessão sobre trabalhos futuros são analisadas outras possibilidades de formatos que talvez ajudem a expandir o público de um sistema como o aqui proposto.

4.1 O Protocolo MIDI

O protocolo *MIDI* (*Musical Instrument Digital Interface*) é uma especificação de um padrão de *software* e *hardware* (MIDI Manufacturers Association, 2012a), usado amplamente na indústria musical para fazer a comunicação entre diferentes instrumentos musicais eletrônicos. *MIDI* foi padronizado por um comitê de representantes da indústria musical, é mantido pela *MIDI Manufacturers Association* e foi introduzido em 1983.

O protocolo define uma série de mensagens digitais (MIDI Manufacturers Association, 2012b) para representar comandos como, por exemplo: notas musicais, mudança de timbres, controle de volume, vibrato, notação temporal e ainda mensagens de sincronização de relógio para que diferentes dispositivos consigam trabalhar em conjunto. Estas mensa-

Software	Formatos de Notação	Formatos de Áudio
FL Studio	MIDI, Notação proprietária, REX	AIFF, DS, DWP, MP3, OGG, SF2, SYN, WAV
Ableton Live 8	MIDI, Max	WAV, AIFF, MP3, ogg, FLAC
PreSonus Studio One	MIDI, Notação Proprietária	WAV, AIFF, REX, CAF, OGG, FLAC, MP3
Apple Logic 9	MIDI, Notação proprietária, GarageBand	WAV, AIFF, MP3, SDII, CAF
Propeller Head Reason 6.5	MIDI, Notação proprietária	WAV, AIFF, SF2
Renoise	MIDI, OSC, XM, IT, MOD	AIF, WAV, FLAC, AIFC, AU, SND, CAF, OGG, MP3, MP4, AAC
Cockos Reaper	MIDI, REX	AIFF, DS, MP3, OGG, SF2, WAV
Cubase Elements 6	MIDI, MusicXML	AIFF, AIFC, WAV, BWAV, MP3, OGG, REX
Avid Pro Tools	MIDI, SIB	WAV, AIFF, MP3, Quick-Time
Cakewalk Sonar X2	MIDI	WAV, MPEG, AIFF, ASF, SND, FLAC, SD2, CAF

Tabela 4.1: Formatos de notação e áudio suportados por DAWs comuns

gens são estritamente comandos e informações sobre parâmetros e não envolvem nenhum som por si só, bem como não definem a qualidade do som a ser produzido pelo instrumento receptor.

O padrão foi estabelecido com o intuito de permitir que um instrumento eletrônico controlasse outro, de tal forma a permitir que sintetizadores fossem desenvolvidos como pequenos módulos apenas responsáveis pela geração sonora e controlados por diferentes interfaces. Além disso permite que mais de um instrumento possa ser controlado simultaneamente pela mesma interface. Isto abriu diversas possibilidades principalmente para performance ao vivo, mas também para composição e produção musical usando estes instrumentos controlados digitalmente. Pode-se dizer que o padrão é extremamente bem sucedido no que é atestado por sua longevidade. Ele é empregado pesadamente ainda hoje e quase todos novos sintetizadores reais ou virtuais são ainda produzidos com suporte ao protocolo.

4.2 Arquivos MIDI, Standard MIDI files

A definição original do protocolo *MIDI* não especifica nenhum tipo de informação de alto nível sobre as mensagens, como por exemplo ordenação temporal, separação em faixas, compassos e melodias. Isto porque a comunicação entre dois instrumentos é dada por uma fluxo de mensagens que implicitamente define a ordem de execução dos comandos (a ordem de chegada) e outras informações semânticas sobre a peça ficam a cargo do músico. Dada esta limitação e com o intuito de capturar uma performance *MIDI* de uma maneira padrão e portátil entre diferentes instrumentos e sistemas foi criado o formato *Standard MIDI File (SMF)*, chamado informalmente de **arquivo MIDI**.

O formato possui informações sobre uma sequência de mensagens MIDI, com um

Formato	Tamanho	Tipo	Compressão
MIDI	52.3K	Notação	Não
WAV	14.8M	Áudio	Não
OGG	1.3M	Áudio	Com perdas
MP3	1.4M	Áudio	Com perdas
FLAC	8.4M	Áudio	Sem perdas

Tabela 4.2: Comparação do tamanho de diferentes formatos de música

tempo relativo ao início da sequência para cada mensagem, bem como informações como sub-divisão em faixas, instrumentação, assinatura de tempo, letras de música e outros metadados. O arquivo final representa uma série de comandos que se executados em um sintetizador irão reproduzir a peça musical nele expressa, similar ao que é uma partitura no sentido mais tradicional de música. Isto significa que arquivos *SMF* serão geralmente muito menores em tamanho de disco do que a mesma música depois de gravada (veja a tabela 4.2 para uma comparação). Além disso arquivos *SMF* são muito mais maleáveis do que arquivos de áudio, pois podem ser editados e renderizados com diferentes sons.

Softwares ou dispositivos de *hardware* que leem arquivos *MIDI* e enviam os comandos nele presentes para um instrumento são popularmente chamados de sequenciadores. Como dito anteriormente, arquivos *MIDI* em si não carregam nenhum áudio, o que implica que não existem garantias de como um arquivo irá soar quando sequenciado em diferentes sintetizadores. Além disso eles são popularmente associados ao áudio de baixa qualidade gerado pelos primeiros sintetizadores em *software* e placas de áudio presentes nos computadores de uso pessoal, mesmo que injustamente.

5 ANALOG FARM

Neste capítulo será apresentado o serviço Analog Farm, com um sistema que implementa os conceitos apresentados neste trabalho. Será discutida em detalhes a sua arquitetura e o seu funcionamento. Além disso serão apresentados alguns casos de uso típico do sistema.

5.1 Uso típico do sistema

Para melhor entender o funcionamento do *Analog Farm* cabe aqui apresentar a utilidade esperada do sistema. O caso de uso típico seria numa produção musical tradicional que emprega faixas de áudio e *MIDI* em um software *DAW* juntamente com instrumentos virtuais e *plugins*. Digamos que o produtor julgue que uma das faixas *MIDI* esteja insatisfatória e decida que gostaria de gravá-la um sintetizador analógico, visto que diversos músicos os consideram como tendo um som que se destaca nas mixagens (PINCH; REINECKE, 2009). Ele pode então acessar o sistema, ouvir amostras dos timbres disponíveis e depois submeter algumas tarefas usando os arquivos *MIDI* de sua gravação. Dado um custo suficientemente baixo de acesso, este processo pode ser repetido diversas vezes até pelos músicos mais humildes. Mesmo com alguns dos problemas inerentes ao acesso remoto comentados nos capítulos anteriores, produtores que sabem o som que desejam e estão acostumados a trabalhar em peças com partes em áudio e partes em *MIDI* podem preparar suas produções usando sons sintetizados por computador e posteriormente submeterem suas composições para gravação pelo Analog Farm.

A preocupação maior do sistema então é permitir que os usuários possam visualizar os sintetizadores disponíveis, registrar tarefas e requisitar gravações da maneira que melhor se integre no seu fluxo de trabalho. As capacidades do sistema percebidas pelo ponto de vista do usuário são:

- Fazer um novo registro de usuário
- Fazer a autenticação como usuário do sistema
- Visualizar os sintetizadores disponíveis
- Ouvir demonstrações dos sons dos sintetizadores disponíveis
- Fazer o upload de uma nova música em formato MIDI
- Requisitar a gravação de uma música com um sintetizador e alguns parâmetros
- Fazer o *download* de gravações

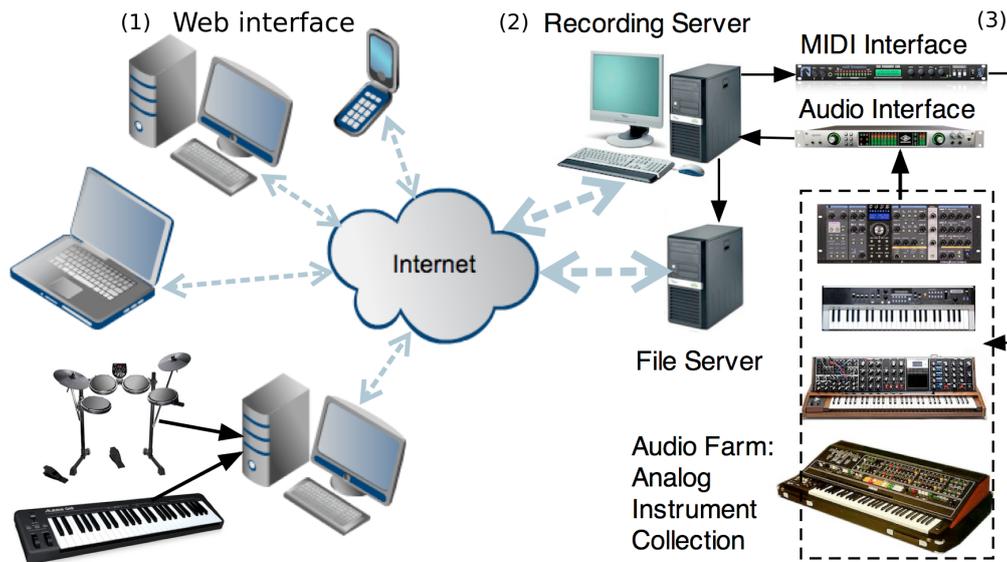


Figura 5.1: Diagrama da Arquitetura do Analog Farm

- Listar suas gravações já prontas

5.2 O Serviço

O *Analog Farm* consiste em um interface *web* e um *webservice REST* (BATTLE; BENSON, 2008) e permite ao usuário acesso a dois diferentes sintetizadores analógicos, um *Roland MKS-50* e um *Studio Electronics ATC-1*. Ele está disponível no endereço <http://www.analogfarm.com> desde o dia 11 de Setembro de 2012 e vem sendo testado por um conjunto de mais ou menos 15 usuários, com um nível variado de experiência com música.

O *Analog Farm* recebe o *upload* de arquivos *MIDI* descrevendo como a música deve ser tocada, isto é referido no contexto do sistema como um “tarefa” enviada pelo usuário. Cada tarefa específica em qual sintetizador ela deve ser tocada, bem como os parâmetros de som a serem usados pelo sintetizador. Uma *URL* com o resultado final é apresentada ao usuário logo que a gravação fica pronta. Múltiplas gravações podem ser requisitadas para cada tarefa a qual o usuário tenha criado e todas ficam acessíveis posteriormente ao processo de gravação.

O sistema basicamente é dividido em três partes: a interface *web* para interação com o usuário (1), o servidor para processamento de tarefas e distribuição de arquivos (2) e os sintetizadores (3), um esquema pode ser visto no diagrama 5.1.

Em alto nível o sistema interage com usuários por meio da interface *web* a qual recebe arquivos de notação musical, permite a escolha de um sintetizador e seus parâmetros e cria uma tarefa de gravação. Esta interface conecta-se ao servidor de gravação, enfileira todas as tarefas dos usuários, processa-as uma a uma e gera os arquivos de áudio resultantes das gravações. Estas gravações são depositadas no servidor de arquivos e enviados como um endereço para os usuários tão logo estiverem prontas. Nos primeiros protótipos os servidores *web*, de arquivos, bem como o de sintetizadores estavam implementados em um mesmo computador, mas como será explicado mais adiante isto não é requerido e uma organização distribuída é desejável.

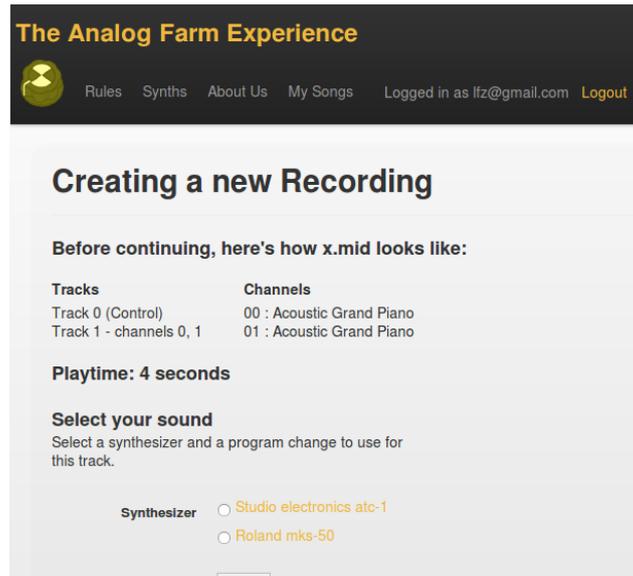


Figura 5.2: Tela de criação de uma nova gravação

5.3 Interface WEB

O primeiro protótipo era muito simples e consistia de um *script* PHP que apresentava um formulário de *upload* de arquivos MIDI. Estes eram colocados em uma fila e processados um por um, passando por um processo de modificação do *MIDI* um pouco mais primitivo do que o atual e sendo encaminhados para um outro *script* que controlava o *DAW*, fazia a gravação e depositava os arquivos resultantes em um endereço que era mostrado ao usuário imediatamente após a submissão da tarefa.

A versão mais atual do sistema foi construída utilizando o framework *Ruby on Rails*, um banco de dados para os usuários, tarefas e gravações e um sistema de filas chamado *delayed_job* (LÜTKE et al., 2009). O sistema está hospedado em um *MacMini* rodando um servidor *Apache* e um banco de dados *PostgreSQL*. A interface gráfica web é bastante simples e usa *HTML5* e *Javascript* padrão para permitir o seu uso num maior número de dispositivos como computadores pessoais, celulares e *tablets*. O formulário de *upload* toma alguns cuidados com o tipo e tamanho máximo permitido dos arquivos e depois da inserção da tarefa permite a visualização de vários parâmetros úteis para o usuário, como as faixas presentes no arquivo, os instrumentos programados em cada faixa e o tempo total de reprodução (figura 5.2). A tela principal do *site* mostra uma listagem de todas as tarefas submetidas por um usuário e para cada tarefa as suas respectivas gravações (figura 5.3), que podem ser visualizadas e obtidas individualmente (figura 5.4).

5.4 Interface Webservice REST

O *Analog Farm* também expõe um interface para acesso como *webservice* respondendo a requisições *HTTP*. Esta interface é trivialmente implementada baseada na arquitetura *REST* e *Ruby on Rails*. Para criar uma nova tarefa um usuário pode fazer uma requisição do tipo *POST* com um arquivo MIDI como conteúdo e irá receber um arquivo *XML* ou *JSON* com um *status* de sucesso e um endereço para acesso dessa tarefa. Uma requisição *POST* para este último endereço com os parâmetros do sintetizador cria uma nova gravação e retorna como resposta um endereço onde ela pode ser consultada e outro

The Analog Farm Experience

Rules Synths About Us My Songs Logged in as lfz@gmail.com Logout

My Songs

	Filename	Recordings	Playtime
Record this Job	single.mid	#1 #2	2.14 minutes
Record this Job	arp.mid		2.48 minutes
Record this Job	x.mid	#1	4 seconds

Upload a new song

Filename Browse...

Upload

Figura 5.3: Tela de listagem de músicas e gravações

The Analog Farm Experience

Rules Synths About Us My Songs Logged in as lfz@gmail.com Logout

Recording #1 of single.mid

Filename single.mid
 Synthesizer Studio Electronics ATC-1
 Program Change 1
 Tracks 1 'Track 1'
 Recorded yet? Yes

File Format

Flac *Compressed lossless audio format, recommended* Size (82.69 Kb)

Wav *Raw audio format, use if your environment has no support for .flac* Size (344.57 Kb)

Zip *Compressed .wav file in zip format*

Download Back

Figura 5.4: Tela de obtenção de gravação

HTTP	Parâmetros	Retorno	Descrição
POST /songs	Arquivo MIDI	Status e endereço da tarefa	Cria uma nova tarefa e recebe um caminho para ela
POST /songs/id	ID do sintetizador e parâmetros de timbre	Status e endereço da gravação	Requisita uma nova gravação de uma tarefa
GET /recordings/id	-	Status da gravação e endereço para download	Consulta informações sobre a gravação e checa se já é possível obter o arquivo
GET /recordings/id/download	Formato de download	Um arquivo no formato especificado	Faz o download do arquivo gravado

Tabela 5.1: Especificação do Webservice REST do Analog Farm

onde pode ser feito o *download*. Requisições posteriores ao endereço da gravação irão retornar um *status* que sinaliza se ela já está pronta ou não e em caso positivo o endereço de *download* pode ser usado para recuperar a gravação. Para autenticação é usado um método de chave de *API*, similar a serviços como *Twitter* e *Facebook*.

A tabela 5.1 mostra um resumo da interface do *webservice*. Futuramente será divulgada uma especificação mais formal desta interface e também alguns exemplos de clientes que fazem uso dela. Um exemplo interessante seria um *plugin* para um *DAW* que pode integrar o uso do *Analog Farm* com suas funções de básicas, fazendo assim o uso do sistema possível sem quebrar o fluxo de trabalho do usuário tendo que acessar uma interface *web*.

6 DETALHES DE IMPLEMENTAÇÃO

Neste capítulo serão explicados alguns detalhes da implementação do *Analog Farm* e discutidas soluções para problemas técnicos encontrados. Em 6.1 pode ser vista uma figura da arquitetura do sistema explicitando as tecnologias utilizadas para implementação e como estão relacionadas.

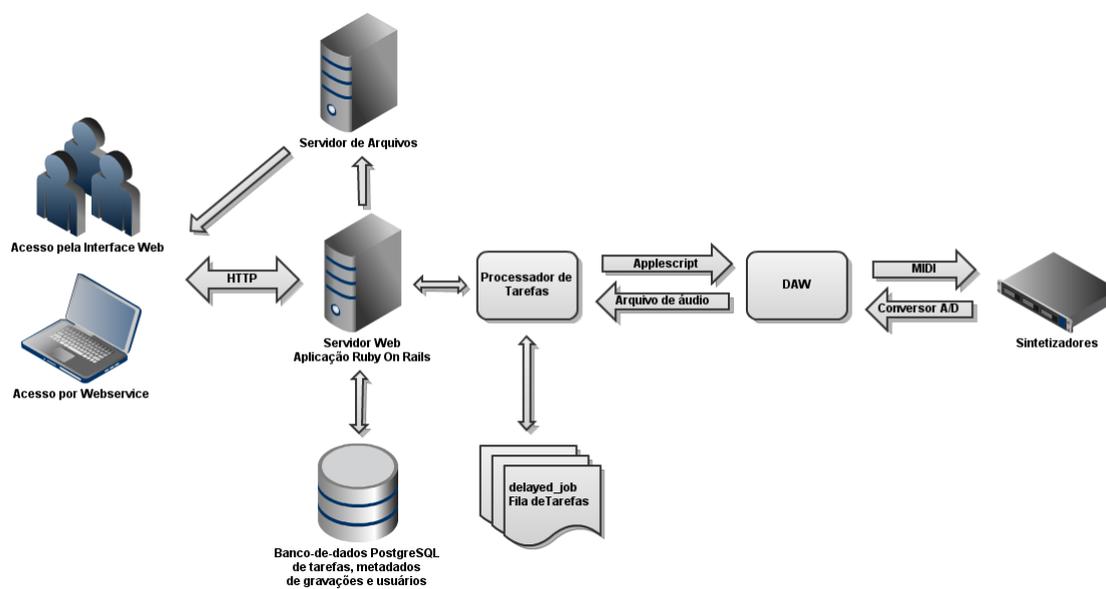


Figura 6.1: Arquitetura de Software do Analog Farm

6.1 Banco de Dados

Os dados pertinentes ao serviço são guardados em um banco de dados *PostgreSQL* hospedado no mesmo servidor que a interface *web*. O esquema do banco e as relações entre as tabelas podem ser visto na figura 6.2. São guardadas informações sobre tarefas (*Jobs*), gravações (*Recordings*), amostras sonoras (*Samples*), usuários (*Users*) e Sintetizadores (*Synthesizers*). O diagrama foi gerado com a ajuda da ferramenta *Umbrello* (HENSGEN, 2007) e os dados foram retirados diretamente do que está sendo utilizado no código de aplicação.

6.2 Tarefas

No contexto da implementação as composições que são enviadas para o sistema por usuários são designadas como *tarefas* e consistem do arquivo *MIDI* e algumas informações

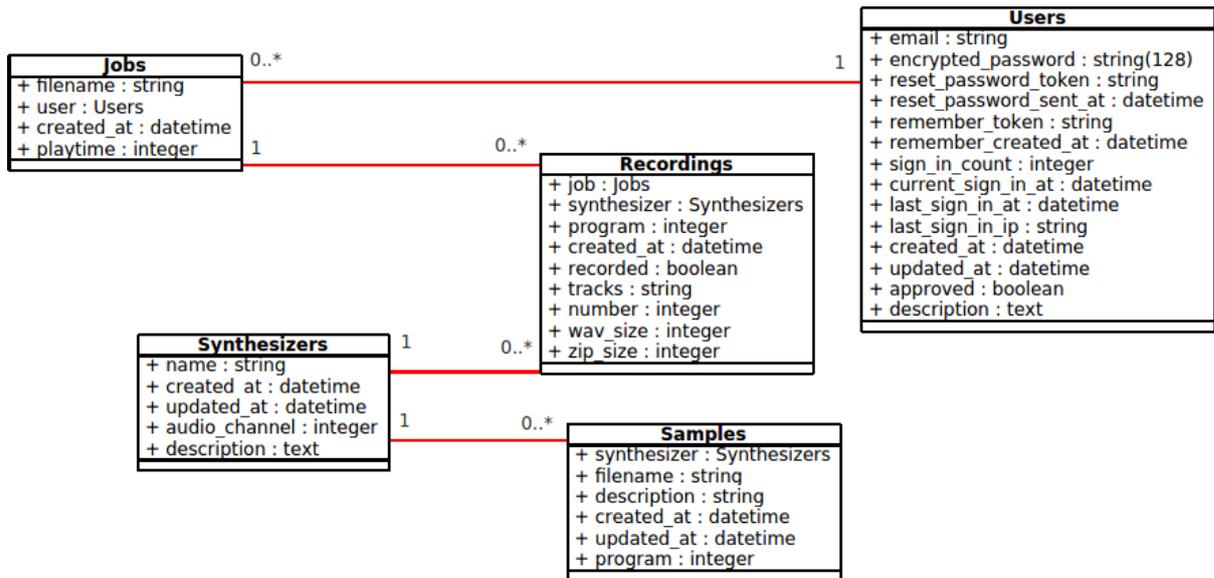


Figura 6.2: Esquema do Banco de Dados do Analog Farm

básicas como o tempo total de reprodução e o usuário associado. Após serem inseridas no sistema todas as tarefas passam por um processamento que adequa-as ao processo de gravação. Este processamento é feito sobre os arquivos *MIDI* e consiste em:

- Cálculo do tempo total que o arquivo necessita para tocar: Para fins do processo de gravação é preciso pré-calcular o tempo que cada faixa *MIDI* gasta para ser tocada. Existe um limite máximo de tempo que cada faixa pode usar, na implementação atual ele é de 10 minutos.
- Separação das diferentes faixas do arquivo *MIDI*: Geralmente arquivos *MIDI* estão organizados em faixas com partes separadas da composição como, por exemplo uma separação da melodia principal e do acompanhamento. Tipicamente o usuário irá querer gravar cada uma dessas faixas com diferentes sintetizadores ou parâmetros, logo elas são separadas para dar ao usuário a opção de escolher quais deseja gravar.
- Remoção da programação de parâmetros: Algumas mensagens *MIDI* são dedicadas a programar parâmetros como o volume, modulação e efeitos. Entretanto o mapeamento da mensagem e sua modificação no som varia dependendo do sintetizador e geralmente causa resultados que o usuário não consegue prever, portanto na versão atual do sistema elas são removidas por padrão.
- Remoção de mensagens *SysEX*: Mensagens *SysEx*, significando *System Exclusive*, são uma parte do protocolo *MIDI* que permite aos implementadores de sintetizadores (ou qualquer sistema que consuma mensagens *MIDI*) definir algumas mensagens com significados especiais e que funcionam exclusivamente para determinado sistema. Elas geralmente são utilizadas para programar parâmetros e funções inusitados ou únicos para alguns sintetizadores. Além disso algumas tratam de comandos que modificam as configurações e os parâmetros globais do sintetizador, então estas são sempre removidas por questão de segurança.
- Normalização dos instrumentos: Além dos parâmetros citados anteriormente, existem algumas mensagens *MIDI* usadas para programar o timbre usado pelo sintetiza-

dor (por exemplo as mensagens *program change* e *bank change*) e geralmente estas não têm relação com o som final que o usuário espera gravar usando o sistema, visto que são todas relativas ao contexto da *DAW* do usuário. Desta forma todas mensagens deste tipo são removidas e depois reconfiguradas pela própria interface *web*.

Também existe uma opção de permitir ao usuário a gravação da tarefa sem nenhuma modificação, salvo a remoção de algumas mensagens de programação e *SysEx*. A ideia por trás disto é a possibilidade de que o próprio usuário tenha construído seu arquivo *MIDI* com as mensagens de programação e seleção de timbres desejadas para a gravação. Entretanto considera-se que, sem utilizar o próprio sintetizador localmente, é muito difícil que o usuário consiga manipular estes parâmetros a ponto de ter certeza de como serão refletidos na gravação. Logo, a decisão preferida ao projetar o sistema foi deixar este tipo de configuração para a segunda parte do processo, dentro da interface *web*, onde o sistema tem um melhor controle sobre como programar o sintetizador.

6.3 Gravações

Quando um usuário requisita uma nova gravação ele deve escolher uma *faixa* do arquivo para ser gravada, um *sintetizador* e um som deste sintetizador. Estas informações são guardadas no banco de dados e é enfileirada uma nova requisição de gravação no sistema de filas.

As gravações são feitas com a ajuda do *DAW Cubase 6 Elements*. Ele é usado para fazer o sequenciamento e comunicação *MIDI* com os sintetizadores, fazer a gravação da entrada analógica e automatizar o processo de corte do conteúdo sonoro. Por si só o Cubase é um *software* monolítico e altamente orientado para usuários de sua interface gráfica o que não encoraja o seu uso num ambiente de servidor onde ele será comandado programaticamente. Entretanto estas operações são automatizadas por um pequeno *script* escrito usando a linguagem *Applescript* (COOK, 2007), uma ferramenta nativa do sistema operacional *MacOS* criada especialmente para o propósito de automatização de interação com a interface gráfica.

O *script* apenas recebe o nome do arquivo *MIDI* e alguns parâmetros de configuração e se encarrega de executar o *DAW*, configurar uma sessão de gravação, esperar o tempo necessário e fechar o programa. A programação com *Applescript* não foi tão trivial quanto se esperava e para conseguir a automatização necessária foi necessário um trabalho de programação da interface do *DAW* para que ele respondesse a diversos atalhos de teclado. Dito isto, o comportamento do *script* é totalmente desacoplado do sistema, de tal sorte que eventuais modificações dele ou a sua total substituição seriam tarefas triviais do ponto de vista de integração com o resto dos módulos.

6.3.1 Obtenção de gravações

Todas as gravações podem ser consultadas por seus respectivos usuários via interface *web* e ficarão listadas lá quando disponíveis para *download*. Para que o usuário possa melhor integrar o sistema com seu ritmo e ambiente de trabalho é possível escolher entre diversos formato de arquivo de áudio para receber a gravação. Tipicamente em processos de gravação áudio é armazenado em arquivos *wav*, um formato muito simples que representa o som como uma série de números. A simplicidade do formato faz com que ele seja unanimemente suportado em *softwares* de gravação e edição de áudio (veja tabela 4.1) e

naturalmente ele é uma opção para os arquivos do Analog Farm.

A desvantagem do formato *wav* é o grande espaço ocupado em disco e em resposta a essa deficiência existem vários formatos de áudio comprimido que podem ser utilizados. Porém note que o nicho que este sistema atende é o da produção musical com equipamentos de alta qualidade então usar formatos de compressão com perda de qualidade (*lossy*), como por exemplo *mp3* e *ogg*, não é desejável mesmo que eles consigam diminuições tremendas do tamanho do arquivo. Portanto o formato usado internamente para armazenar as gravações, e o recomendado para o *download* dos usuários, é *flac* (COALSON, 2012). Este é um formato de áudio com compressão e sem perdas de qualidade que ajuda na economia de espaço. Entretanto ele ainda hoje não é tão bem suportado em diferentes *softwares* como o *wav*, mesmo conseguindo um fator de compressão de em média 50-60% do tamanho do arquivo original (COALSON, 2012).

Dito isto os formatos suportados para *download* são: *flac*, *wav*, arquivo *wav* comprimido em formato *zip*. Este último não é uma compressão tão boa quanto *flac*, por não ser específica para áudio, mas é um bom compromisso também para usuários que necessitam de arquivos *wav* e gostariam de economizar tempo de transferência. Além disto foi implementado um pequeno tocador na página de cada gravação onde é possível escutar uma versão de baixa resolução em formato *mp3/ogg* para se ter uma prévia antes de tentar consultar os arquivos de verdade. Na tabela 4.2 é possível ver uma comparação do tamanho dos diversos formatos para um mesmo arquivo.

6.3.2 Hardware de gravação

Os sintetizadores são conectados ao servidor usando uma placa *Texas PCM4222EVM* adaptada com os *drivers* de entrada feitos sob medida (JOHANN, 2010). As gravações são feitas com uma qualidade de 24bits e amostragem de 48KHZ usando cada sintetizador conectado a um canal diferente da interface de gravação. Para a comunicação de mensagens MIDI está sendo usada uma interface *MIDI USB MOTU TimePiece*. Na figura 6.3 pode ser visto o servidor do Analog Farm. Em 6.4 podem ser vistos os sintetizadores e a interface gravação, de cima para baixo temos: o *Roland MKS-50* a interface de gravação e o *Studio Electronics ATC-1*.



Figura 6.3: Servidor onde está hospedado o Analog Farm



Figura 6.4: Interface de gravação, Roland MKS-50, Fonte de Alimentação e Studio Electronics ATC-1

6.4 Distribuição de arquivos

Um problema decorrente da infraestrutura atual do sistema é que o servidor onde estão conectados os sintetizadores, o servidor onde as gravações são geradas e o servidor da interface *web* estão conectados à *internet* por uma conexão com velocidade de *upload* não muito alta. Por decorrência disto os *downloads* de vários usuários simultâneos acabam por deixar a interface *web* bastante sobrecarregada e com dificuldade de manter a qualidade de serviço para outros usuários que estão apenas navegando pelas páginas do *site*. Sem uma mudança estrutural, mesmo com uma conexão com maior banda este problema existiria dado um número significativo de acessos.

Uma solução para este problema já foi esboçada na descrição alto-nível dada no início deste capítulo e ela consiste no uso de um servidor diferente para fazer a distribuição de arquivos. Infelizmente é preciso pelo menos um primeiro *download* do arquivo de gravação para transferi-lo do servidor conectado aos sintetizadores para algum outro e isto decorre em possivelmente mais tempo de espera para o usuário. Uma troca que pode ser feita é começar a transferência para outro servidor assim que a gravação estiver pronta e, caso o usuário requira o arquivo antes dele estar no servidor de arquivos, usar o servidor de gravação para mandar este arquivo. Ainda é necessária uma melhor análise do perfil dos usuários e das requisições que fazem dos arquivos para ter certeza de que isto realmente aumentaria a disponibilidade das gravações.

6.5 Automatização de Implantação, Testes e Monitoramento

O sistema *Analog Farm* se mostrou bastante complexo de implantar e manter num ambiente de produção. Isto decorre do fato de existirem diversos pontos onde alguma falha pode ocorrer e geralmente os erros decorrentes não serem percebidos facilmente, ou pior, serem apenas percebidos pelos usuários. Falhas no sintetizador, no *DAW*, no processamento *MIDI* e nas demais configurações de gravação geralmente só eram notadas quando o usuário recebia um arquivo defeituoso como resultado da gravação. Entre os defeitos comuns no período de testes pode-se citar gravações sem nenhum conteúdo, gravações com o volume muito baixo e algumas vezes gravações com resultados inesperados (leia-se, não refletindo a música que o usuário requisitou para gravação).

Falhas no *software* que serve a interface *web* são um pouco mais simples de diagnosticar, visto que o servidor pode guardar registros de todas as requisições para ele, detectar ocorrências de erros e disparar alertas para o administrador se necessário. Outra preocupação além da monitoração da execução do sistema era o processo de implantação de novas versões do código, geralmente correlacionado com a descoberta de novos defeitos inesperados.

Neste capítulo serão explicadas as técnicas de automatização de implantação de novas versões do código, bem como métodos de teste e monitoramento do *software* rodando em servidor.

6.5.1 Implantação do Software

Uma das situações mais temidas por desenvolvedores é serem obrigados a lidar com as inconsistências do ambiente de desenvolvimento e do ambiente de produção. Diferentes versões de compiladores, interpretadores e bancos de dados são garantia de problemas quando chega o momento de implantar o código e há quem diga (WIGGINS, 2012) que é preciso trabalhar ativamente para que se consigam reduzir estas diferenças ao mínimo possível. Entretanto no caso do *Analog Farm* existe pouca esperança de conseguir diminuir o abismo entre os dois ambientes (pelo menos não da maneira convencional, na sessão sobre testes analisa-se possibilidades para tentar melhorar a situação).

Para começar, o servidor onde o software deve ser implantado tem um perfil de *hardware* diferente do computador típico usado apenas para programação, mas este não é um problema tão sério. O que dificulta realmente a situação é que o maior recurso do qual o sistema faz uso, os sintetizadores, não estão presentes no ambiente de desenvolvimento e testes, pois estão conectados apenas ao servidor remoto para uso com o sistema. Este fato sozinho complicou imensamente a etapa de implantação de código e adicionou várias sub-etapas de teste *ad hoc* de interação pela interface *web*, procurando por defeitos que poderiam ser causados por falhas nas camadas inferiores como o sintetizador, configurações de gravação, *MIDI*, e etc. . .

Para tentar diminuir esta dificuldade, além do uso de testes, o protocolo de implantação foi automatizado com a ajuda de um conjunto de ferramentas de software. A principal delas é o sistema de versionamento distribuído de código *Git* (TORVALDS, 2005) e os seus *hooks*. O código do *Analog Farm* desde o início do desenvolvimento estava versionado dentro de um repositório *Git*, que é atualizado com novas versões constantemente na máquina local de desenvolvimento e eventualmente com novas versões do software final no servidor. Sendo assim, inicialmente, o processo de atualização e implantação do código era feito com o uso do comando *git push*, significando “transmita as mudanças locais para o repositório no servidor”. Infelizmente era difícil evitar uma posterior configuração manual dos sintetizadores e reinicialização do servidor *web* e sistema de filas. Isto tornava o processo bastante demorado e estressante, principalmente quando as modificações necessárias eram apenas pequenas correções, desencorajando um ritmo de desenvolvimento mais fluído e potencialmente aumentando o tempo que o sistema fica inacessível.

Neste ponto que entra a utilidade dos *hooks* do *Git* e sua contribuição para o processo de implantação do *Analog Farm*. Os *hooks* são *scripts* que podem ser executados pelo *Git* antes ou após algumas das ações, como por exemplo: o preparo de um *commit*, a recepção uma nova versão do código ou a aplicação de um patch. Este sistema foi colocado em prática primeiramente como um *hook* do lado do servidor que ao receber uma nova versão do código, executava o seguinte processo automaticamente:

- Espera o término do processamento da próxima tarefa na fila
- Para o funcionamento do servidor web
- Para o funcionamento do sistema de filas
- Aplica as modificações do código
- Reinicia o ambiente de software
- Instala novas dependências
- Faz modificações necessárias no banco de dados
- Reinicia o servidor web, as filas e o DAW (se necessário)

Este processo geralmente dura alguns segundos e não prejudica o tempo de acesso do sistema para os usuários, além de facilitar a implantação de novas versões rapidamente. Entretanto ele tem problemas, visto que os testes pós-implantação ainda precisam ser feitos manualmente. Além disso é desejável que as operações ajam como se fossem uma transação, para evitar que o repositório acabe em um estado não funcional e desconhecido no caso de algum passo que falha. Essa característica não existe por padrão no *Git* e portanto essa organização ainda é muito vulnerável a essas falhas, podendo causar períodos de inatividade do sistema e trabalho manual para o desenvolvedor.

Uma solução para estes problemas de implantação foi a inclusão de uma versão de “teste” do sistema, rodando no servidor de produção, mas acessível apenas para os administradores. Essa versão usa a mesma infra-estrutura de sintetizadores e filas (tarefas do administrador tendo prioridade sobre aquelas atualmente na fila) e então é possível detectar manualmente alguns defeitos mais comuns que acontecem na hora de implantação e mesmo assim manter o serviço funcionando usando a versão antiga. Além disso, qualquer erro que ocorra durante o próprio processo de implantação pode ser revertido facilmente na versão de teste e estas mudanças só são repassadas para a versão final quando se tem muito mais certeza de que irão funcionar. Este processo de implantação em “duas fases” permite algumas facilidades que serão expandidas quando discutido na sessão sobre testes de validação.

6.5.2 Monitoramento

Erros na interface *web* significam para o usuário telas de erro *HTTP* número 500, indicando um erro interno do servidor ou telas de erro 404, indicando página não encontrada (no caso do usuário receber esta mensagem tentando acessar uma página válida). No caso do *Analog Farm* são geralmente causados por erros de programação que desencadeiam falhas em alguma parte do código de aplicação. Como dito anteriormente, erros na interface *web* são mais fáceis de descobrir e diagnosticar, visto que cada uma de suas ocorrências podem ser registradas, com direito à coleta informações preciosas de depuração graças ao *framework* de desenvolvimento usado. No momento cada um dos erros que são registrados na aplicação é coletado num banco de dados separado e avisado para um desenvolvedor por *e-mail*.

Para facilitar o processo de depuração do processo de gravação foi também criado uma interface de monitoramento das filas. Com ela é possível visualizar todas as tarefas pendentes, parar e iniciar as filas e analisar erros que aconteceram em cada tarefa.

Originalmente, era comum que algumas tarefas, após serem gravadas, demonstrassem defeitos e não refletissem o resultado esperado pelo usuário. Como solução para os casos mais simples foi implementada um teste sobre o resultado da gravação, uma checagem da presença do arquivo no lugar esperado, e assim determinava-se se o processo havia sucedido ou não. Em caso negativo a tarefa seria reinserida na fila para ser processada novamente mais tarde. Infelizmente esta abordagem não se mostrou muito útil quando o erro era recorrente, pois incorria uma tarefa sendo reinserida na fila diversas vezes sem nenhum sucesso, talvez por decorrência de algum defeito nas configurações dos servidores ou dos sintetizadores.

O método adotado no momento consiste em tentar refazer apenas mais uma vez a tarefa em caso de falha na gravação. Caso haja reincidência da falha, serão notificados os administradores e a tarefa será pausada e sinalizada como em estado de erro. Então pode ser usada a interface de monitoramento para obter maior informação sobre o erro e reabilitar a tarefa após a correção. A interface também se encarrega de guardar estatísticas como: número de tarefas processados, número de tarefas que acabaram em falha e tempo médio por tarefa.

6.5.3 Painel de Controle

Da necessidade de centralizar o diagnóstico de erros e o comando de outras funções úteis do sistema nasceu o painel de controle. Nele podem ser reiniciados e monitorados o servidor *web*, o sistema de filas e o *DAW*. Além disso ele mostra informações úteis como um agregado dos erros que aconteceram na aplicação, estado do disco do servidor de arquivos, informações de tráfego recolhidas com a ajuda do serviço *Google Analytics* e integra a tela de monitoramento das filas.

6.5.4 Testes Unitários, de Integração e de Validação

A prática de testes automatizados de software, popularmente associada à Programação Orientada a Testes (*TDD* do inglês *Test Driven Development*), é uma estratégia para diminuir a complexidade dos projetos de software, aumentar a qualidade do código e facilitar o processo de manutenção. O *Analog Farm* foi projetado com esta filosofia em mente, mas a aplicação de testes no processo de desenvolvimento e implantação se mostrou desafiadora em alguns aspectos.

Para o código da lógica de aplicação, como por exemplo o processamento e conversão de arquivos, foram usados testes unitários implementados com a ajuda das bibliotecas embutidas no *framework Ruby on Rails*. O mesmo pode ser dito da interface *web* e dos aspectos de *webservice* do sistema que foram testados por meio de requisições *HTTP* e simulação de navegação por um usuário. Estes testes ajudam muito no período de desenvolvimento e assegurando com algum grau de certeza a correção do funcionamento do sistema.

O desafio fica caracterizado pela etapa de testes de integração, onde é preciso executar a fila de tarefas e de alguma maneira simular o acontecimento das gravações, mesmo no ambiente de desenvolvimento. O compromisso encontrado foi fazer o *mocking* do processo de gravação, isto significa, criar mecanismos que simulam o comportamento real do ambiente de produção. Desta forma no ambiente de desenvolvimento as tarefas quando executadas são vistas como um processo de espera de tempo seguido da criação de um arquivo de áudio como se sucede após o decorrer do processo real de gravação. Isto significa que uma tarefa de gravação de uma música de 5 minutos também irá durar 5 minutos no ambiente de desenvolvimento e os sistemas de monitoramento e detecção de

falhas na gravação podem ser auditados até certo ponto sem precisar recorrer ao ambiente de produção.

Durante um período inicial do desenvolvimento o arquivo de áudio criado por esse processo *mock* era apenas um arquivo vazio que simbolicamente representava o sucesso da gravação. Depois resolveu-se refinar um pouco mais este “truque” para que ele pudesse ser usado como um teste um pouco mais abrangente. Foi implementada uma função do sistema que calcula a duração de tempo do arquivo de gravação gerado (um arquivo no formato *wav*) e este tempo pode ser comparado com o esperado pela tarefa, aproximando um pouco mais o ambiente de produção do ambiente de testes automatizados.

Como explicado na sessão anterior, para fazer um teste de validação usando os sintetizadores de verdade o processo se tornava mais complicado e isto levou ao desenvolvimento do ambiente de testes que pode ser executado lado a lado com o sistema em produção. Neste ambiente também pode-se realizar testes automatizados e uma ideia é comparar gravações de uma mesma tarefa pré-definida, a qual se tem uma noção do que esperar. Nota-se que é inútil comparar *bit a bit* duas gravações de uma mesma tarefa, pois pequenas diferenças no processo de gravação irão tornar os arquivos diferentes, mas é totalmente possível testar se parâmetros como a frequência, volume médio e duração de tempo estão dentro de intervalos esperados. Este tipo de teste é realizado com tarefas representando apenas um nota musical, durante um período de alguns segundos e ajuda a assegurar que depois da implantação o sistema esteja funcionando de uma maneira satisfatória.

Ao integrar estes testes ao processo de implantação em duas fases definido anteriormente o desenvolvedor pode ter muito mais certeza de que após o lançamento de uma nova versão do código o sistema irá realizar suas funções livre de defeitos. Estes testes também podem ser acionados a partir das interfaces de monitoramento para averiguar a existência ou não de problemas com o funcionamento e configuração do serviço.

6.5.5 Testes de Carga

Foram realizados de forma bastante primitiva alguns testes de carga do sistema. Milhares de tarefas foram criadas programaticamente a fim de obter amostras sonoras dos sintetizadores, que seriam disponibilizadas posteriormente por meio da interface *web* como parte do processo de escolha do sintetizador pelo usuário. Os testes tiveram sucesso e ficou comprovado que sob circunstâncias normais de acesso o sistema de enfileiramento de tarefas funciona bem e consegue atender os usuários mesmo na presença de mais tarefas do que os sintetizadores podem atender imediatamente.

6.5.6 Testes da Interface REST

Os testes do comportamento da interface *REST* foram realizados de maneira *ad hoc* e em parte automática juntamente com os testes da interface *web*. No futuro espera-se fazer uma automatização destes testes usando iniciativas como o *REST-unit* (BORGES; PIMENTA; WEBER, 2009) e permitir uma validação do sistema já em produção, do ponto de visto de um consumidor do *webservice*.

6.6 Desafios técnicos e Soluções

6.6.1 Diferenças entre um servidor tradicional e um servidor de gravações

Desde muito cedo, ao construir a Analog Farm, havia um dilema a ser resolvido na hora de escolher a máquina que hospedaria o serviço. O fato é que para servir requisições *web*, fazer a transferência de arquivos para clientes e aguentar o tráfego de navegação chegamos a um perfil comum de *hardware* e *software*. Este tipo de aplicação geralmente se beneficia de um servidor com boa concorrência de processos, grande quantidade de memória *RAM* e boa velocidade de leitura de disco. O perfil de um computador usado para gravações não é necessariamente este, no caso da tarefa de gravar sintetizadores analógicos, as prioridades são apenas garantir a fidelidade do envio de comandos pela interface *MIDI* e prevenir atrasos no processamento de áudio, visto que estes podem causar defeitos audíveis no resultado final.

No fim das contas tudo que se quer é manter a precisão temporal do controle e gravação do sintetizador o mais alta possível. Este processo é simples o bastante com hardware *COTS* moderno e sob condições de uso normal, por exemplo, um usuário gravando o sintetizador conectado ao seu computador. Infelizmente começa a se tornar mais complexa quando incluída num servidor que estará constantemente lendo e escrevendo no disco, usando grande a memória *RAM* para criar processos que servem cada cliente e escalonando estes processos juntamente àquele que está fazendo a gravação. A perspectiva do processo de gravação sendo preemptado do uso do processador não é nada atraente e os erros evasivos causados por problemas de performance como este são extremamente difíceis de detectar e corrigir.

Assim é caracterizado o dilema do *Analog Farm* que tem que co-hospedar um serviço *web* deste tipo com um servidor que está apenas fazendo o controle e gravação de sintetizadores. Para contribuir para esta situação adversa, o perfil de *hardware* e *software* requerido para gravar áudio profissionalmente nem sempre combina com o descrito anteriormente como o ideal para agir como servidor *web*. Por exemplo servidores tipicamente usam sistemas operacionais como *Linux* ou *BSD*, as vezes até mesmo versões modificadas destes. Entretanto fabricantes de *hardware* especializado em áudio dificilmente suportam estas duas plataformas e geralmente têm requisitos bastante específicos. Além disso a necessidade de interação com os sintetizadores torna impossível terceirizar os servidores usando serviços *cloud* como os oferecidos pelas empresas *Google* ou *Amazon*.

6.6.2 Servidor de gravação distribuído

Da mesma forma que foi proposta a separação do servidor de arquivos, pode-se imaginar também a separação dos sintetizadores em diferentes servidores que seriam acessados por um servidor central onde ficaria localizada apenas a interface *web* e o *webservice*. Este servidor seria responsável por autenticar usuários do sistema, enfileirar suas gravações e distribuí-las para os sintetizadores corretos. Note que isto também facilitaria o uso de redundância de sintetizadores e a distribuição de carga de gravações para diferentes instâncias criando uma arquitetura propícia para a escalabilidade.

Um protótipo inicial deste tipo de arquitetura já está funcional e espera-se que a versão final permita que a interface do sistema seja movida para algum serviço de hospedagem baseado em nuvem, de forma a minimizar as preocupações com manutenção de infraestrutura, *backups* de arquivos e confiabilidade.

7 CONCLUSÃO

O objetivo principal deste trabalho é criar um sistema que permita o acesso fácil a sintetizadores analógicos raros e/ou caros para o propósito de obtenção de gravações de alta qualidade. O trabalho demonstra de maneira satisfatória que a implementação deste sistema é possível e que ele é capaz de contribuir para o aumento da qualidade de novas gravações musicais de diversos músicos. Acredita-se que os problemas técnicos atacados nesse trabalho são de interesse dos implementadores de outros sistemas que desejem prover acesso otimizado a recursos caros como é o exemplo dos sintetizadores.

O emprego de processamento em lote e automatização do processo de gravação contribuem para aumentar a taxa de utilização dos sintetizadores, diminuir o custo do serviço e permitir o atendimento de mais usuários do que se acessando o sistema usando algum sistema de agendamento prévio.

Espera-se que o sistema também ajude a preservar a diversidade sonora das várias gerações de sintetizadores ao promover o uso contínuo de equipamentos reais. Muitos sintetizadores que não são capazes de controle *MIDI* podem ser adaptados com trabalho dedicado de *hardware* e acredita-se que este esforço será válido não somente sob o viés econômico, mas também para ajudar a manter instrumentos icônicos vivos e em contato com o trabalho das novas gerações.

7.1 Trabalhos futuros

7.1.1 Instrumentos musicais e equipamentos como servidores

Em (ZAWACKI; JOHANN, 2012a) é discutida de uma maneira prospectiva a ideia de empregar servidores web como “provedores” de acesso a instrumentos remotos, usados para produção sonora e gravação, e dispositivos usados para a modificação de arquivos sonoros pré-gravados. Neste trabalho também são observados os limites entre o que pode ser aproveitado e servido em tal sistema, pelo o ponto de vista de um estúdio de gravação. Uma contribuição importante do trabalho é o estabelecimento de que com as tecnologias de controle padrão, como por exemplo *MIDI* é possível construir um sistema de acesso remoto para diversos instrumentos acústicos, não apenas sintetizadores.

São analisados pianos, instrumentos de cordas, instrumentos de sopro, órgãos e instrumentos de percussão. Outra contribuição importante é definir alguns aspectos de um estúdio que não se adequam muito bem à arquitetura proposta, como por exemplo microfones e pré-amplificadores.

7.1.2 Melhorias no Analog Farm

O *Analog Farm* continuará evoluindo no futuro e a prioridade é a implementação de servidores de gravação de maneira distribuída. Isto irá permitir uma melhor escalabilidade do serviço, maior testabilidade dos ambientes e novas possibilidades não muito exploradas nesse trabalho. Um exemplo desta possibilidades é a instalação do ambiente de gravação em servidores controlados por terceiros, com intuito de que estes compartilhem seus sintetizadores usando parte da infraestrutura já estabelecida pelo *Analog Farm*.

Planeja-se também consolidar uma interface de programação de *webservices* e prover exemplos de clientes deste serviço, no intuito de abrir para outros desenvolvedores a possibilidade de integrarem gravação de sintetizadores analógicos aos seus próprios serviços. Além disso o plano é criar interfaces próprias para *softwares* de gravação e ambientes especializados como tablets e celulares, visando prover uma experiência menos intrusiva no processo de composição dos usuários.

No momento o *Analog Farm* permite em sua interface web a escolha de *patches* pré-programados, nome comum para uma coleção de parâmetros que cria um timbre específico em um sintetizador. Para cada sintetizador a decisão de qual usar pode ser feita baseando-se em amostras pré-gravadas dos sons que estão disponíveis para os usuários. No futuro deseja-se implementar uma interface *web* que permita uma programação mais livre dos sintetizadores ou até mesmo usando arquivos *MIDI* ou *SysEx*. Estes novos *patches* ficariam associados à conta do usuário que os criou e seriam carregados dinamicamente no sintetizador antes da gravação, permitindo assim que os usuários expandam a biblioteca de timbres, criem os sons que desejarem para suas gravações e talvez até compartilhem-os com outros usuários.

Também existem planos para de expandir o número de formatos de notação suportados, baseado na avaliação feita na sessão de Notação Musical. O candidato mais promissor é o MusicXML (GOOD, 2001), mesmo que ele só possua suporte em um dos *DAW* listados, visto que o mesmo pode ser dito da maioria dos outros. Suas vantagens incluem a existência de um grande catálogo de peças livremente disponíveis na *internet*, diversos conversores de ida e volta para outros formatos e a sua representação textual *XML* que abre novas possibilidades para sua manipulação junto a navegadores *web* e interfaces com *webservices*.

REFERÊNCIAS

- BARBOSA, A. Displaced Soundscapes: a survey of network systems for music and sonic art creation. **Leonardo Music Journal**, [S.l.], v.13, p.53–59, Dec. 2003.
- BARRY, S. Chapter XV: and how it grew. In: **Hammond As In Organ: the laurens hammond story**. [S.l.: s.n.], 1974.
- BASET, S.; SCHULZRINNE, H. An analysis of the skype peer-to-peer internet telephony protocol. **arXiv preprint cs/0412017**, [S.l.], 2004.
- BATTLE, R.; BENSON, E. Bridging the semantic Web and Web 2.0 with representational state transfer (REST). ... **Science, Services and Agents on the World Wide Web**, [S.l.], v.6, n.1, p.61–69, Feb. 2008.
- BODE, H. Sound Synthesizer Creates New Musical Effects. **Electronics**, [S.l.], Dec. 1961.
- BODE, H. European Electronic Music Instrument Design. **J. Audio Eng. Soc**, [S.l.], v.9, n.4, p.267–269,304, 1961.
- BORGES, F. B.; PIMENTA, M. S.; WEBER, T. S. REST-Unit: geração baseada em u2tp de drivers de teste para restful web services. In: XXXV LATIN AMERICAN INFORMATICS CONFERENCE, Pelotas, RS. **Anais...** [S.l.: s.n.], 2009. p.11–21.
- BRISTOW-JOHNSON, R. Wavetable synthesis 101, a fundamental perspective. **Proc. AES Convention**, [S.l.], p.1–27, 1996.
- CÁCERES, J.; CHAFE, C. JackTrip: under the hood of an engine for network audio. **Journal of New Music Research**, [S.l.], v.39, n.3, p.183–187, Sept. 2010.
- CARLOS, W. **Switched-on Bach**. [S.l.]: Columbia Records, 1968.
- City of Melbourne. **Town Hall Organ**. Disponível em: <<http://www.melbourne.vic.gov.au/>>. Acesso em: 14 de Dezembro, 2012.
- COALSON, J. **FLAC: free lossless audio codec**. Disponível em: <<http://flac.sourceforge.net/>>. Acesso em: 11 de Dezembro, 2012.
- COOK, P. R. **Sampling Synthesis**. [S.l.]: AK Peters, 2002. 11–18p.
- COOK, W. R. AppleScript. In: ACM SIGPLAN CONFERENCE ON HISTORY OF PROGRAMMING LANGUAGES - HOPL III, New York, New York, USA. **Proceedings...** ACM Press, 2007. p.1–1–1–21.

GOOD, M. MusicXML: an internet-friendly format for sheet music. **XML Conference and Expo**, [S.l.], p.1–12, 2001.

Google Inc. **YouTube Press Statistics**. Disponível em: <http://www.youtube.com/t/press_statistics>. Acesso em: 11 de Dezembro, 2012.

HENSGEN, P. **Umbrello UML Modeller**. Disponível em: <<http://uml.sourceforge.net/>>. Acesso em: 12 de Dezembro, 2012.

JOHANN, M. d. O. **Modifying the ADA8000**. Disponível em: <<http://www.inf.ufrgs.br/johann/ada8000/>>. Acesso em: 12 de Dezembro, 2012.

JORDÁ, S.; BARBOSA, A. Computer supported cooperative music - overview of research work and projects at the Audio Visual Institute–UPF. ... **on Current Research Directions in Computer Music**, [S.l.], 2001.

KLAPURI, A.; DAVY, M. **Signal Processing Methods for Music Transcription**. [S.l.]: Springer, 2006. 440p. v.2.

KORG. **Analog Ribbon Synth Korg Monotrons**. Disponível em: <<http://www.korg.com/monotron>>. Acesso em: 11 de Dezembro, 2012.

LÜTKE, T. et al. **delayed_job**. Disponível em: <https://github.com/collectiveidea/delayed_job>. Acesso em: 16 de Dezembro, 2012.

MAES, L.; RAES, G.; ROGERS, T. The man and machine robot orchestra at logos. **Computer Music Journal**, [S.l.], n.Kapur 2005, 2011.

MARTIN, S. M. et al. **Theremin: an electronic odyssey**. [S.l.]: MGM/UA Home Entertainment, 2001.

MAYTON, B.; JOLIAT, N.; PARADISO, J. A. Patchwerk: multi-user network control of a massive modular synthesizer. **NIME Proceedings**, [S.l.], 2012.

MIDI Manufacturers Association. **General MIDI 1 and 2 Specification**. Disponível em: <<http://www.midi.org/techspecs/gm.php>>. Acesso em: 11 de Dezembro, 2012.

MIDI Manufacturers Association. **MIDI Message Table**. Disponível em: <<http://www.midi.org/techspecs/midimessages.php>>. Acesso em: 14 de Dezembro, 2012.

Moog Music Inc. Subtractive Synthesis Tutorial. In: **Moog Voyager User's Manual**. [S.l.: s.n.], 2012. v.27, n.6, p.3–4.

Moog Music Inc. **Moog Voyager Synthesizer**. Disponível em: <<http://www.moogmusic.com/products/Minimoog-Voyagers>>. Acesso em: 12 de Dezembro, 2012.

MOOG, R.; RHEA, T. Evolution of the keyboard interface: the bösendorfer 290 se recording piano and the moog multiply-touch-sensitive keyboards. **Computer Music Journal**, [S.l.], v.1, n.2, 1990.

PERPETUO, I. F.; SILVEIRA, S. A. da. Os fãs e a cultura participativa nos sites de redes sociais. In: **O Futuro da Música Depois da Morte do CD**. São Paulo: Momento Editorial, 2009. p.101–102.

PIMENTA et al. Ubiquitous Music: concepts and metaphors. **Proceedings of the Brazilian Symposium on Computer Music**, [S.l.], p.139–150, 2009.

PINCH, T.; REINECKE, D. Technostalgia: how old gear lives on in new music. **Sound Souvenirs: Audio Technologies, Memory and Cultural Practices.**, Amsterdam, p.152–166, 2009.

POLYCOM. **Music Performance and Instruction over High-Speed Networks**. [S.l.: s.n.], 2011. (June).

REID, G. An Introduction To Additive Synthesis. **Sound On Sound**, [S.l.], June 2000. Disponível em: <<http://www.soundonsound.com/sos/jun00/articles/synthsec.htm>>. Acesso em: 11 de Dezembro, 2012.

REID, G. An Introduction To Frequency Modulation. **Sound On Sound**, [S.l.], Apr. 2000. Disponível em: <<http://www.soundonsound.com/sos/apr00/articles/synthsecrets.htm>>. Acesso em: 11 de Dezembro, 2012.

REID, G. Arturia MiniBrute. **Sound On Sound**, [S.l.], 2012. Disponível em: <<http://www.soundonsound.com/sos/mar12/articles/arturia-minibrute.htm>>. Acesso em: 11 de Dezembro, 2012.

RENAUD, A.; CARÔT, A.; REBELO, P. Networked music performance: state of the art. **Proceedings of the AES 30th ...**, [S.l.], n.4, p.1–7, 2007.

ROESLER, V. et al. Mconf: an open source multiconference system for web and mobile devices. In: **Multimedia - A Multidisciplinary Approach to Complex Issues**. [S.l.]: ISBN: 978-953-51-0216-8, InTech, 2012.

ROSSUM, D. Making digital filters sound analog. In: INTERNATIONAL COMPUTER MUSIC CONFERENCE. **Proceedings...** [S.l.: s.n.], 1992.

SOUNDCLOUD. **Soundcloud Celebrates Reaching 10 Million Milestone on SoundCloud**. Disponível em: <<https://soundcloud.com/press/releases/2012/01/23/ten-million>>. Acesso em: 11 de Dezembro, 2012.

STEINBERG. Steinberg Virtual Studio Technology Plug-in Specification 2.0 Software Development Kit. **Hamburg: Steinberg Soft-und Hardware GMBH**, [S.l.], p.1–84, 1999.

SYNTHTOPIA. **2012 Synthtopia DAW Reader Poll**. Disponível em: <<http://www.synthtopia.com/content/2012/06/18/10-best-daws-in-the-world/greater>>. Acesso em: 12 de Dezembro, 2012.

SYSOMOS. **Top YouTube Videos and Other Statistics**. Disponível em: <<http://www.sysomos.com/reports/youtube/>>. Acesso em: 11 de Dezembro, 2012.

TORVALDS, L. **Git: source control management**. Disponível em: <<http://git-scm.com/>>. Acesso em: 16 de Dezembro, 2012.

WEIDENAAR, R. Telharmonium - Dynamophone. In: **THE NEW GROVE DICTIONARY OF MUSICAL INSTRUMENTS**. Volume 3.ed. [S.l.]: Macmillan Press Ltd., 1984. p.4–6.

WIGGINS, A. **The Twelve-Factor App X. Dev/prod parity**. Disponível em: <<http://www.12factor.net/dev-prod-parity>>. Acesso em: 11 de Dezembro, 2012.

Yamaha Corporation. **DX7: digital programmable algorithm synthesizer**. Disponível em: <<http://yamaha.com/products/musical-instruments/keyboards/synthesizers/dx7/extgreater>>. Acesso em: 11 de Dezembro, 2012.

ZAWACKI, L. F.; JOHANN, M. O. A Prospective Analysis of Analog Audio Recording with Web Servers. **Cadernos de Informática**, [S.l.], 2012.

ZAWACKI, L. F.; JOHANN, M. O. A System For Recording Analog Synthesizers With The Web. **Proceedings of the International Computer Music Conference**, [S.l.], p.128–131, 2012.