

# Composição Musical Com Dados de Satélite: Desafios ao “Musicar” Números

Gabriel D’Incao<sup>1</sup>, Artemis Moroni<sup>2</sup>

[g253628@dac.unicamp.br](mailto:g253628@dac.unicamp.br), [artemis.moroni@cti.gov.br](mailto:artemis.moroni@cti.gov.br)

<sup>1</sup>Música - Instituto de Artes - Universidade de Campinas - UNICAMP

<sup>2</sup>Divisão de Sistemas Ciberfísicos – DISCF  
CTI/MCTI Renato Archer – Campinas/SP

**Abstract.** *This article explores the creative process and challenges faced in composing music using data, specifically satellite data. The author describes the context of the GaiaSenses project, in which they were responsible for the compositions, and analyzes their results, seeking to understand the variation from artistic approaches to more technical ones that made up their learning. Finally, there's a description of what will be done in the future within the project and a synthesis of what was observed with the different patches, considering the advancements in electronic compositions and the main changes that were identified from one patch to another.*

**Resumo.** *Este artigo explora o processo criativo e os desafios encontrados no processo de compor música utilizando dados, em específico, dados de satélite. O autor descreve o contexto do projeto GaiaSenses, no qual ficou encarregado de fazer as composições, e analisa seus resultados buscando entender a variação de abordagens artísticas para outras mais técnicas, que compuseram seu aprendizado. Por fim, há uma descrição do que será feito futuramente dentro do projeto, e uma síntese do que foi observado com os diferentes patches, considerando os avanços das composições eletrônicas e as principais mudanças que foram identificadas de um patch para o outro.*

## 1. Introdução

Vivemos uma era transitória. Estamos no mundo real, mas recorremos ao universo digital para quase todas as nossas ações concretas. Temos uma vida totalmente independente da realidade, que criamos como cópias imperfeitas da nossa, nas redes sociais. Essas vidas são geridas por um algoritmo complexo baseado em dados que doamos inconscientemente, que nos influencia diretamente todos os dias. Penso que, em breve, nossa realidade material será inteiramente transferida a essa outra, digital, e serviremos apenas como um gabinete para assegurar esse novo mundo.

Acredito que qualquer forma de existência é digna, mas gostaria que a nossa existência física e o mundo vivo voltassem a ser importantes, pois possuem intensa profundidade, e devem ser valorizados. Penso que seja possível conectar as pessoas ao mundo novamente, através da arte, e essa ideia é o motor que me guia à fazer música.

Creio que o projeto GaiaSenses (Moroni, 2022; Moroni et al., 2022) está se encaminhando para se tornar uma ótima plataforma de alívio e conexão com a natureza. Nele, temos o objetivo de criar um aplicativo de celular, e um site, onde um usuário poderá checar as condições climáticas de sua posição geográfica, junto à uma composição audiovisual gerada em tempo real, que usa dados de satélite como base para sua geração. Isso conecta o usuário com o mundo à sua volta, através do universo digital, que inicialmente o desconectou.

Neste artigo, explorarei minhas dificuldades ao ter que traduzir o mundo dos dados para algo que fosse capaz de conectar os ouvintes através do áudio. Mostrarei exemplos sonoros e visuais para ilustrar o processo criativo e as dificuldades enfrentadas. Por fim, indicarei quais serão meus próximos passos dentro do projeto.

## **2. Metodologia**

Este capítulo descreve o método utilizado para a sonorização dos dados de satélite. Farei isso explicando um pouco das funcionalidades básicas da linguagem e ambiente escolhido para gerar as composições sonoras (Pure Data), e como ele é capaz de gerar sons através da parametrização de entradas numéricas, contexto necessário para o entendimento do resto do artigo. Começarei explicando o que é este programa (FARNELL, 2010).

### **2.1 Instrumento Musical**

Gosto de pensar o Pure Data, ou "Pd", como um instrumento musical. Assim como em um instrumento "acústico", como o violão, o Pd utiliza da manipulação de um material para movimentar moléculas de ar, produzindo som. No violão, tocamos na corda para que ela vibre, e no Pd, damos instruções de como o alto falante deve se movimentar. Ambos geram o mesmo fenômeno acústico, e não há diferenças qualitativas no resultado sonoro de cada um, eles diferem apenas em sua matéria.

Porém, argumento que essa diferenciação material é superficial. O violão é feito de madeira, cordas, que podem ser de nylon ou aço, e outros materiais de fácil acesso. O computador é feito de alumínio, silício, fibra de vidro, plástico, dentre outros materiais mais raros, que, montados de forma correta, geram um sistema capaz de muitas coisas, inclusive emitir sons. Ambos instrumentos são confeccionados a partir de materiais tangíveis, originários do mesmo planeta, portanto, a diferença física de um violão e um computador está em sua complexidade, funcionalidade, e forma de controle.

Os dois, porém, são instrumentos musicais, e sua única diferença qualitativa real está no modo com que interagimos com cada um. No violão, nossas interações são com cordas, que respondem imediatamente aos nossos estímulos. No computador, há um atraso, que varia de tamanho dependendo do programa sendo utilizado. E isso implica em uma relação fria do instrumentista com seu instrumento. Só após uma conexão mais profunda com o programa, que é possível se expressar com intimidade.



Nesse exemplo, temos um patch, que manda uma série de informações para a caixa de som do seu dispositivo, aqui disposto de maneira lógica e compreensível. Vamos começar pelo X, no meio do mapa. Ao clicar nesse objeto, é iniciado o processo de criação e transmissão de mensagens. Uma informação de "vá" (1) substitui a anterior "não vá" (0). Essa mensagem é então passada por um fio de telefone ao objeto abaixo dele, que desempenha outras funções. Assim que recebida a mensagem para começar, ele começa a desempenhar sua função: mandar pulsos repetidos constantemente à um ritmo indicado pelo comando à sua direita. O [metro 100] está desempenhando a função de mandar "1" ou "bang" a cada 100 milissegundos, valor que poderia ser alterado se houvesse uma mensagem recebida no seu segundo "inlet" (entrada para cabos). Esses comandos rápidos chegam a diferentes departamentos do caos, onde, sempre que ordenados, produzem um número aleatório e mandam abaixo através do cabo (objeto "random"). Esses valores são comandados a entrar nos cabos e mandar mensagens para outros objetos, que podem ter a função de gerar valores constantemente ("osc"), criar envelopes sonoros ("vline~"), ou criar cópias de sons, criando a ilusão de um ambiente tridimensional ("reverb03": um objeto que eu criei usando as ferramentas do Pd).

Os sinais são processados por múltiplos departamentos de matemática, onde a informação é moldada à vontade do arquiteto da cidadela, até que chegam à caixa de som, que traduz esses dados a movimentos rápidos e amplos, gerando um fenômeno acústico percebido por nosso sistema auditivo. Desenhar dentro desse universo é devagar, e apresenta um leque quase infinito de possibilidades sonoras. É necessário muito controle para conquistar algo com valor artístico, principalmente quando deve-se usar, de forma significativa, dados estáticos gerados por satélites. Isso torna difícil o processo de criação, pois balancear valor estético e coerência dos números parece contraintuitivo.

### **3. Os Patches**

Iniciei minha jornada no projeto em um momento no qual eu tinha pouca experiência com o Pd, cerca de 6 meses de contato. Eu havia composto apenas alguns patches dos quais enxergava como contendo potencial artístico, mas ainda não entendia o programa com propriedade. Tinha, porém, um amplo contato com o mundo dos sintetizadores modulares analógicos, que possuem grande semelhança com o processo utilizado pelo Pd.

#### **3.1 ShockWind**

A primeira composição feita para o GaiaSenses foi uma performance de um patch que simulava artisticamente o interior de uma nuvem emitindo trovões. Nesse programa, explorei a ideia de utilizar subpatches que são como funções, nas quais você pode encapsular patches inteiros em um único objeto, de forma que o programa fique mais organizado e facilite o entendimento e a capacidade de fazer "debugs". Fiz diversas cópias de um objeto que filtra ruídos de forma dinâmica, e alguns outros sons, que possuem certa autonomia. Eu modifiquei alguns dos parâmetros do patch, passando por

um reverb externo, e o resultado é algo com um bom valor artístico, mas sem interatividade com os dados de satélite (Figura 2, Figura 3):

<https://drive.google.com/file/d/1X4iHeshQZH9dqH8yUG-jQZRzKPkgVTR0/view?usp=sharing>

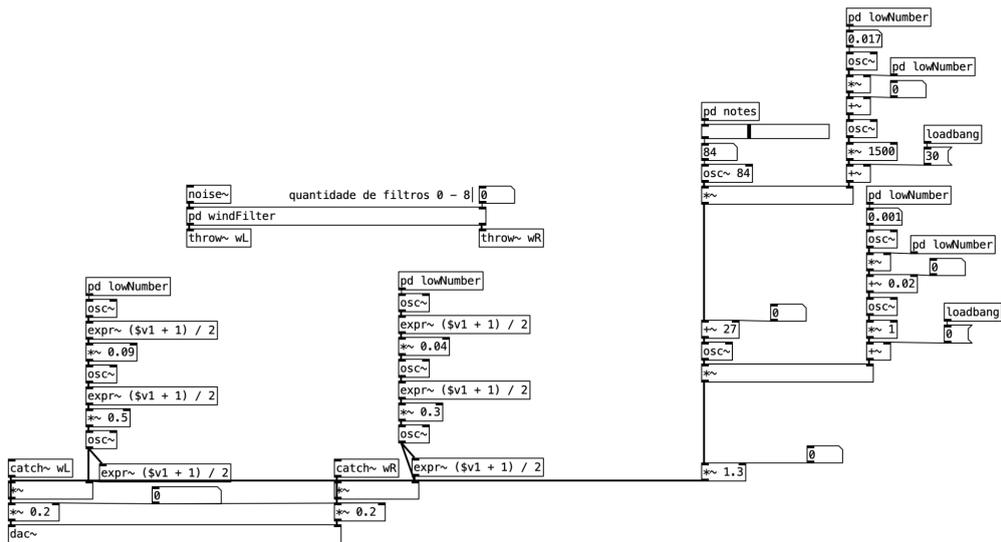
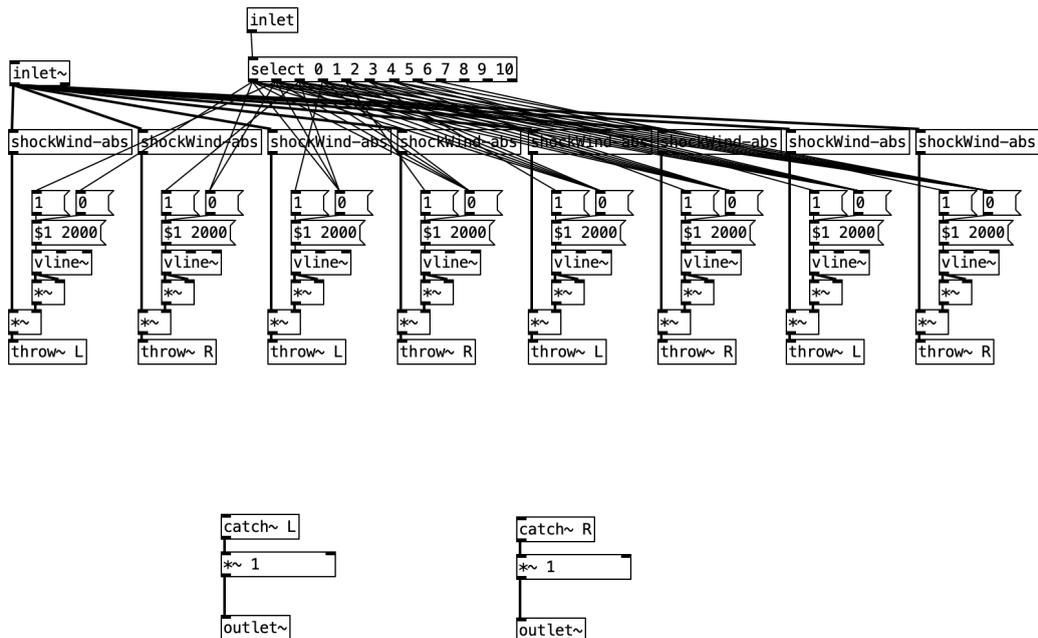
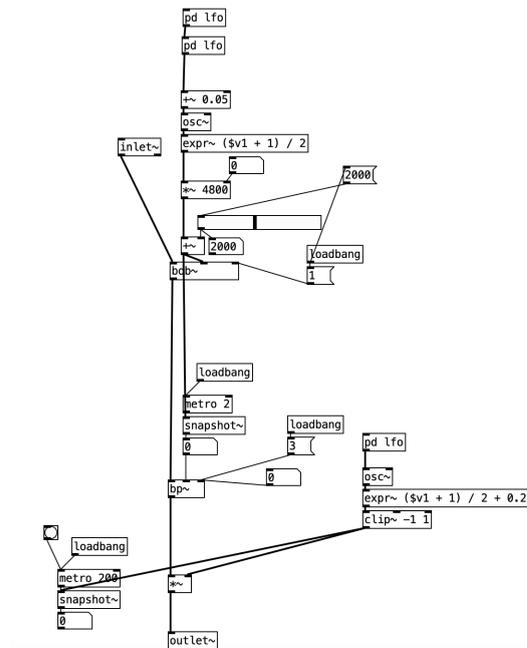


Figura 2: Primeira Camada do Patch. Nela, temos um ruído "noise~" passando por um subpatch "pd WindFilter", que filtra o ruído com um número variável de filtros. Temos também alguns sons graves que simulam os raios.



**Figura 3: Segunda camada do Patch. Um algoritmo seleciona a quantidade de filtros, que variam fluidamente.**



**Figura 4: Terceira camada do patch. Dentro do subpatch "shockWind-abs", uma abstração que varia o valor de frequência de corte de um filtro tipo band pass "bp~" de forma fluida e generativa.**

As escolhas artísticas foram inspiradas por uma animação gerada anteriormente no grupo que utilizou dados de satélite de todo o Brasil para ilustrar a ocorrência de raios em um determinado momento, criando um visual com raios azuis se movimentando em um retângulo (Rigue & Moroni, 2022). O feedback gerado por essa composição foi positivo, e houve um incentivo à utilização dos dados de satélite por meus colegas e orientadora, pois é o real desafio e objetivo dessa área de estudo.

### 3.2 Gotas

A próxima composição baseou-se na animação Gotas, na qual gotas coloridas surgiam aleatoriamente na tela, a uma velocidade coerente com o valor de milimetragem de chuva (Costa & Moroni, 2023). Compus então um patch simples (Figura 5) que simula sons de gotas caindo na água, com uma entrada para variar a velocidade na qual os sons ocorriam. Esse patch foi, após algumas tentativas, implementado dentro do programa da animação de forma que o patch e animação funcionam em conjunto, usando o "libPd" (BRINKMANN et al., 2011). Esse foi um avanço importante para aprendermos

como funciona a implementação do Pd no programa que será responsável por gerar as animações e funcionamentos do aplicativo.

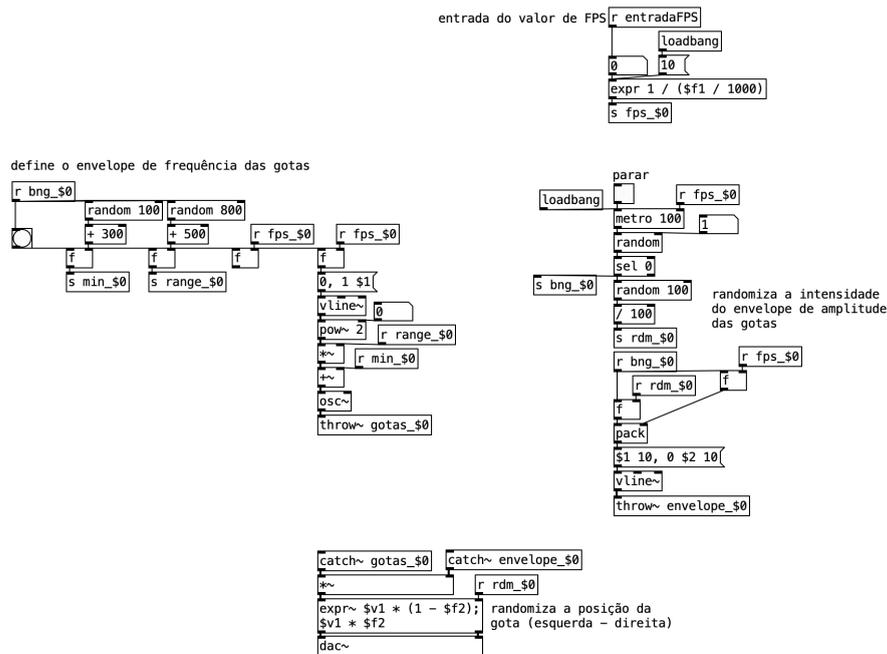


Figura 5: Patch gotas

<https://drive.google.com/file/d/1bx2eCtsOMKDQYInv-HsZnLW43K9Xhll3/view?usp=sharing>

Conseguimos, portanto, implementar pela primeira vez dados de satélite dentro do Pd: O valor de milimetragem de chuva era utilizado para gerar uma animação no processing, valor esse que manipulava a velocidade na qual pequenos círculos eram gerados aleatoriamente na tela. Esse valor de FPS era, então, mandado para o Pd (utilizando o objeto [r entradaFPS]) e traduzido para valores em milissegundos (com a expressão [expr 1 / (\$f1 / 1000)]), que descreviam a velocidade na qual o efeito sonoro de gotas deveria ser gerado. Entendemos então, como poderia funcionar a implementação do Pd no Javascript, e estávamos prontos para compor outras animações mais interessantes, e seguimos adiante.

### 3.3 Zigzag

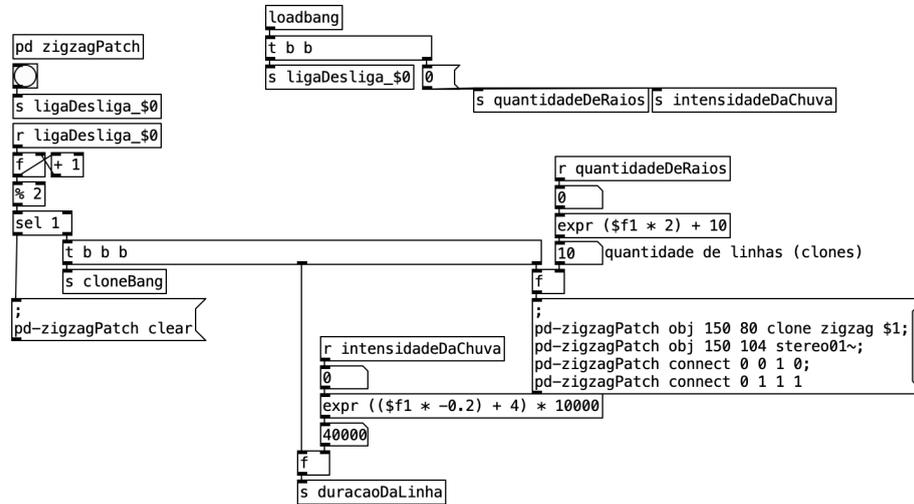
Fui instruído, então, a fazer a sonorização de uma outra animação, que utilizava outros parâmetros de satélite para sua geração. Tentei, depois de algumas tentativas, representar quase exatamente as informações visuais transmitidas pelo vídeo generativo, e o resultado ficou interessante, assista:

<https://drive.google.com/file/d/1SipM-KAdj1YLytdBgKJp2ni7Ddmm6U8H/view?usp=sharing>

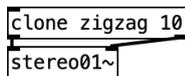
Nesse vídeo, foi usado o objeto [clone] para criar várias instâncias de um patch que possui um algoritmo de randomização timbrística e movimentação fluida, similar à animação, a qual tentei recriar. Cada uma das cópias da melodia timbrística (objeto [zigzag]) tentava sonorizar cada linha, individualmente. Fiz isso observando como elas se comportavam de acordo com as entradas de dados, e simulei isso com a soma de diversos controles estilo LFO, que controlavam diversos parâmetros do timbre e da frequência do som, para assim representar os pequenos movimentos em *zigzag* das linhas descendentes. Após algumas reuniões com o designer, foram recriados os detalhes no patch de forma que a quantidade de raios registrada na região do usuário) alterasse a quantidade de cópias daquele som, que é como a animação funcionava também. A seguir, o link para a animação em conjunto do programa:

<https://drive.google.com/file/d/1-ZlQ-P0JuTJpgZHf0QC07fdJ4KNs5sZz/view?usp=sharing>

Nesse patch (Figura 6), utilizei o mesmo objeto e algoritmo - com algumas alterações mínimas - para que pudesse usar a entrada dos dados para influenciar a quantidade de clones sonoros. Fiz isso utilizando uma técnica chamada "dynamic patching", onde, através de comandos por mensagem, podemos criar objetos descritos em um formato de texto, que pode receber variáveis. Assim, criei uma plataforma de forma que os dados poderiam ser inseridos de forma correta. Os dados mandam a mensagem de criar um objeto [clone zigzag \$1], onde \$1 é a quantidade de cópias do patch "zigzag" que devem ser feitas dentro do subpatch [pd zigzagPatch] (Figura 7). Dessa forma, encontrei uma solução para que a quantidade de clones mude, sem que seja necessário uma interação direta com o objeto via mouse e teclado.



**Figura 6: ZigZag (Dynamic Patching)**



**Figura 7: clone zigzag \$1 (dentro do subpatch [pd zigzagPatch])**

Esse patch foi especialmente útil por conta do conhecimento adquirido com a sua realização. O dynamic patching foi uma importante descoberta, pois serve como solução para um dos problemas que prevemos ser relevante no futuro, que é o tamanho e peso dos patches no CPU da máquina na qual o aplicativo estaria funcionando, já que nos abriu possibilidades de escrever todo o patch em um arquivo de texto, que pode ser acionado com o comando de uma mensagem. Dessa forma, o patch fica bem mais leve, e não precisa ser simplificado. Aprendemos e criamos, com esse processo, uma forma de implementar meus objetos customizados no Javascript com o LibPd, instalando uma biblioteca exclusiva do nosso projeto nos arquivos do programa. Mas não estava satisfeito com o resultado estético, então me propus a fazer outro melhor.

### 3.4 Zigzag 2

Voltei a me preocupar um pouco com a expressão artística, e deixei a abordagem técnica um pouco de lado. Mentalizei e organizei uma ideia sonora que dialogava com a

composição visual e com a ideia das chuvas e raios, os quais representei descritivamente e subjetivamente (Figura 8).

Fiz uma demonstração sonora do patch, onde brinco um pouco com os parâmetros de entrada, que deveriam ser substituídos pelos dados de satélite: [https://drive.google.com/file/d/1McJXTfOCnD9c6RX\\_SZzj-7ya6TSYG21d/view?usp=sharing](https://drive.google.com/file/d/1McJXTfOCnD9c6RX_SZzj-7ya6TSYG21d/view?usp=sharing)

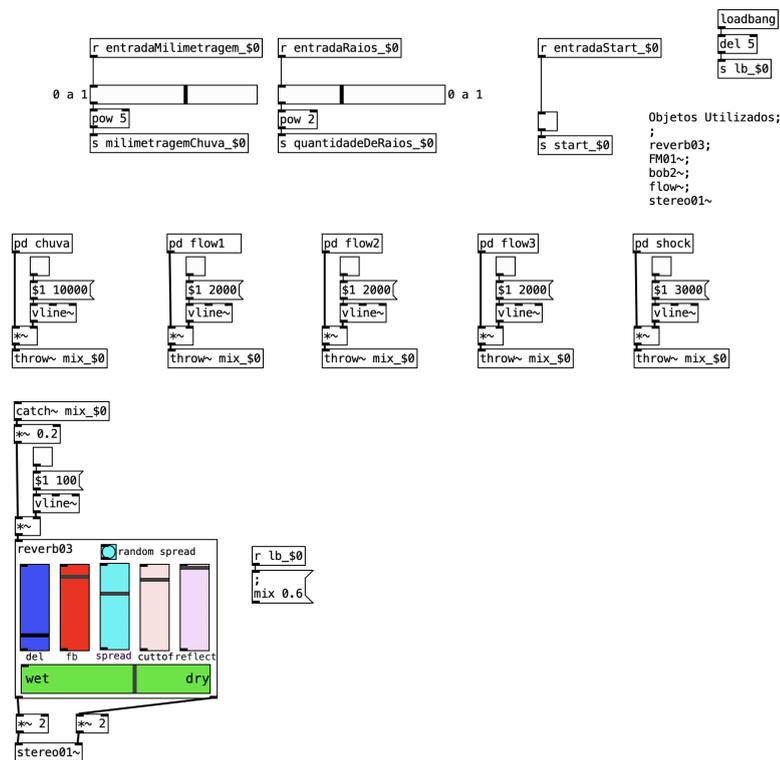


Figura 8: ZigZag 2

O resultado sonoro deste patch me agradou, e os dados de controle alteravam significativamente a intensidade dos objetos sonoros. As representações diretas de chuva e raios faziam sentido, e não eram totalmente óbvias. Os sons "flow" nos ambientavam no universo das nuvens elétricas, e fomos bem sucedidos na implementação do programa dentro do Javascript, colocando os objetos meus que eram necessários para o funcionamento do patch.

Darei uma maior atenção ao objeto [flow~], pois ele é o que caracteriza o som do patch (Figura 9):

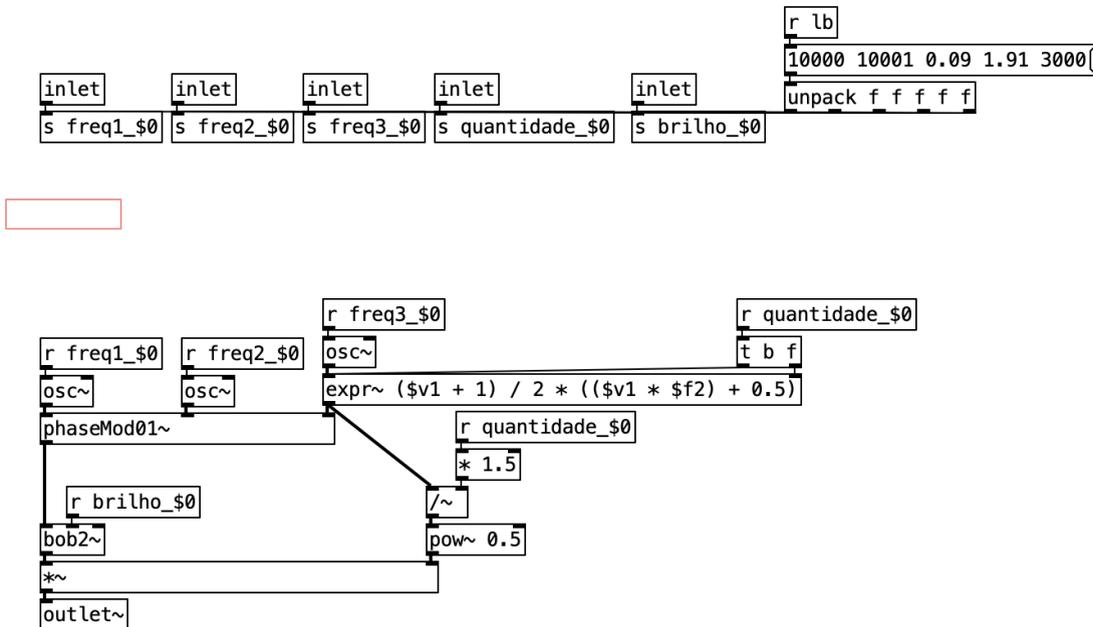


Figura 9: objeto [flow~]

Este objeto controla os parâmetros de um outro objeto meu: [phaseMod01~], que modula a fase de um sinal por outro, com controle sobre a quantidade de modulação, também por sinal. Fiz um preset das frequências utilizadas para os osciladores, assim obtendo um resultado previsível, mas com algumas alterações. Cada subpatch [pd flow] possui um desses objetos com seus próprios valores de controle, gerando objetos sonoros distintos que expressam uma sensação timbrística generativa.

#### 4. Conclusão e Próximos Passos

Tentei explicitar, através deste artigo, meu processo criativo dentro do projeto até então, focando no contraponto dos diferentes patches em suas abordagens, e evidenciando o desenvolvimento técnico/artístico. Iniciei de forma "livre", indicando como imagino a estética ideal do aplicativo, passei por uma fase de aprendizado, onde tive de dialogar com os meus colegas para tornar possível a interação da imagem com o som, então elaborei dois patches que simulavam essa sincronia audiovisual. Em seguida, parti para uma abordagem "equilibrada" entre esses dois opostos, com o ZigZag 2.

Encontrar esse equilíbrio não é uma tarefa fácil, e acredito que ainda devo experimentar e aprender bastante para desenvolver mais patches dentro do projeto, de forma coerente e interessante. É importante, também, o aprendizado sobre as formas de diálogo multidisciplinar, algo essencial em projetos como o GaiaSenses. Foi interessante para mim observar como fomos capazes de conversar sobre linguagens muito diferentes com certa fluência, que veio com a prática. Estamos aprendendo, em grupo, com nossos

experimentos, e aos poucos estamos criando obras audiovisuais que ilustram nossos aprendizados e nos aproximam de implementações práticas com alto valor artístico.

Para o futuro, me imagino continuando este procedimento, onde faço a sonorização das obras audiovisuais de meus colegas, e tento implementar as interações com os dados de satélite. Tenho algumas ideias sonoras para algumas animações que estão sendo desenvolvidas no projeto, e imagino que terei pelo menos dois outros patches finalizados até o fim de 2023.

## Referências

BRINKMANN, Peter et al. Embedding pure data with libpd. In: PROCEEDINGS OF THE PURE DATA CONVENTION, 291., 2011, Citeseer, 2011. Disponível em: <<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=bdabb0fb166a71d426a962346920170cb45cc66eL>>. Acesso em: 29/05/2023

Costa, A., Moroni, A. GaiaSenses: o Frontend e Backend de uma aplicação móvel para a geração automática de composições audiovisuais baseadas em dados climáticos. In: JICC2023 - Jornada de Iniciação Científica do CTI Renato Archer, 2023, Campinas.

Farnell, A. (2010) Designing Sound. London: The MIT Press

MORONI, A. Data Art, Criatividade Computacional e Mudanças Climáticas. In: SIIMI2022 - Simpósio Internacional de Mídias Interativas, 2022, Santiago, Chile. Anais do Simpósio Internacional de Inovação em Mídias Interativas. Goiânia: Media Lab/UFG, 2022. , 2022.

MORONI, A.; RAMOS, J.; MANZOLLI, J.; COLTRI, P.; GONCALVES, R.; ZULLO, J.; BAINY, B.; VELLOSO, B.; CARDOSO, A.; MARTINS, P.; DEZOTTI, C. G.; KUAE, G. T.; Tauane Leme Monteiro Cardoso; Elton Cardoso do Nascimento1; LACERDA, T. D. S.; RIGUE, I.; SHEN, R. GaiaSenses: mobile application for generating audiovisual compositions from planetary platforms In: Data Art for Climate Action, 2022, Hong Kong. Proceeding|Catalogue of Data Art for Climate Action (DACA2022). Hong Kong: School of Creative Media, 2022. p.140 - 141

RIGUE, I.; MORONI, ARTEMIS GaiaSenses: Acesso à base de dados de satélite e tratamento de seus produtos In: JICC2022 - Jornada de Iniciação Científica do CTI Renato Archer, 2022, Campinas. JICC2022. CAMPINAS: CTI Renato Archer, 2022.

XXV Jornada de Iniciação Científica do Centro de Tecnologia da Informação Renato Archer - JICC 2023  
PIBIC/CNPq/CTI - Outubro de 2023 – Campinas – São Paulo