

# Simulador de eletromagnetismo na guitarra usando p5.js

Clóvis de Almeida Pelosini<sup>1</sup>

Centro Universitário Senac - Campus Santo Amaro

## Introdução

Sempre gostei muito de tocar música, des de muito pequeno, toco guitarra e conforme o tempo foi passando ficava mais curioso para saber como que o instrumento funciona, como eu toco uma corda e ele sai algum som para o cabo, um dia acabei perguntando para um professor de guitarra e ele me explicou o básico do funcionamento, assim pensando que seria uma ótima ideia transformar isso em algo visível e interativo, mostrando a força do eletromagnetismo.

## Metodologia

A metodologia que usei foi primeiro pensar em qual o problema/ideia que teria que resolver e o melhor jeito de se executa-la, após pensar bem, passei a ideia para código sempre começando com partes simples e depois deixando-as mais complexas. Quando tenho alguma dúvida não gosto de pesquisar antes de tentar algo da minha cabeça, acredito que isso ajude bastante no processo do aprendizado.

## Desenvolvimento

Primeiramente, pensei em representar o funcionamento da simulação. Comecei testando com uma corda da guitarra para simular a vibração e sua frequência, utilizando a seguinte função trigonométrica:

$$f(x) = amp \cdot \sin((x + mov) \cdot freq \cdot T) \quad (1)$$

Onde **amp** é a amplitude da onda, **x** é o ângulo (ou posição no eixo X), **mov** é o deslocamento para criar movimento, e **freq** é a frequência multiplicada por **T**, uma constante temporal. Com essa fórmula, simulei uma corda e depois adicionei outras cinco cordas fixas. Criei um algoritmo para identificar a corda selecionada pelo clique do usuário, transformando-a em uma onda. Após implementar a seleção e ajuste das cordas com sliders, pensei na identificação das notas pela frequência. Encontrei tabelas como a disponível em **Muted.io Note Frequencies**.

Observando as tabelas, percebi que não era eficiente armazenar todas as frequências no p5. Notei que a frequência de uma nota em oitavas diferentes é um múltiplo de 2 da frequência base. Assim, a frequência de uma nota pode ser calculada por:

$$noteFreq \cdot (2^{\text{oct}-1}) \quad (2)$$

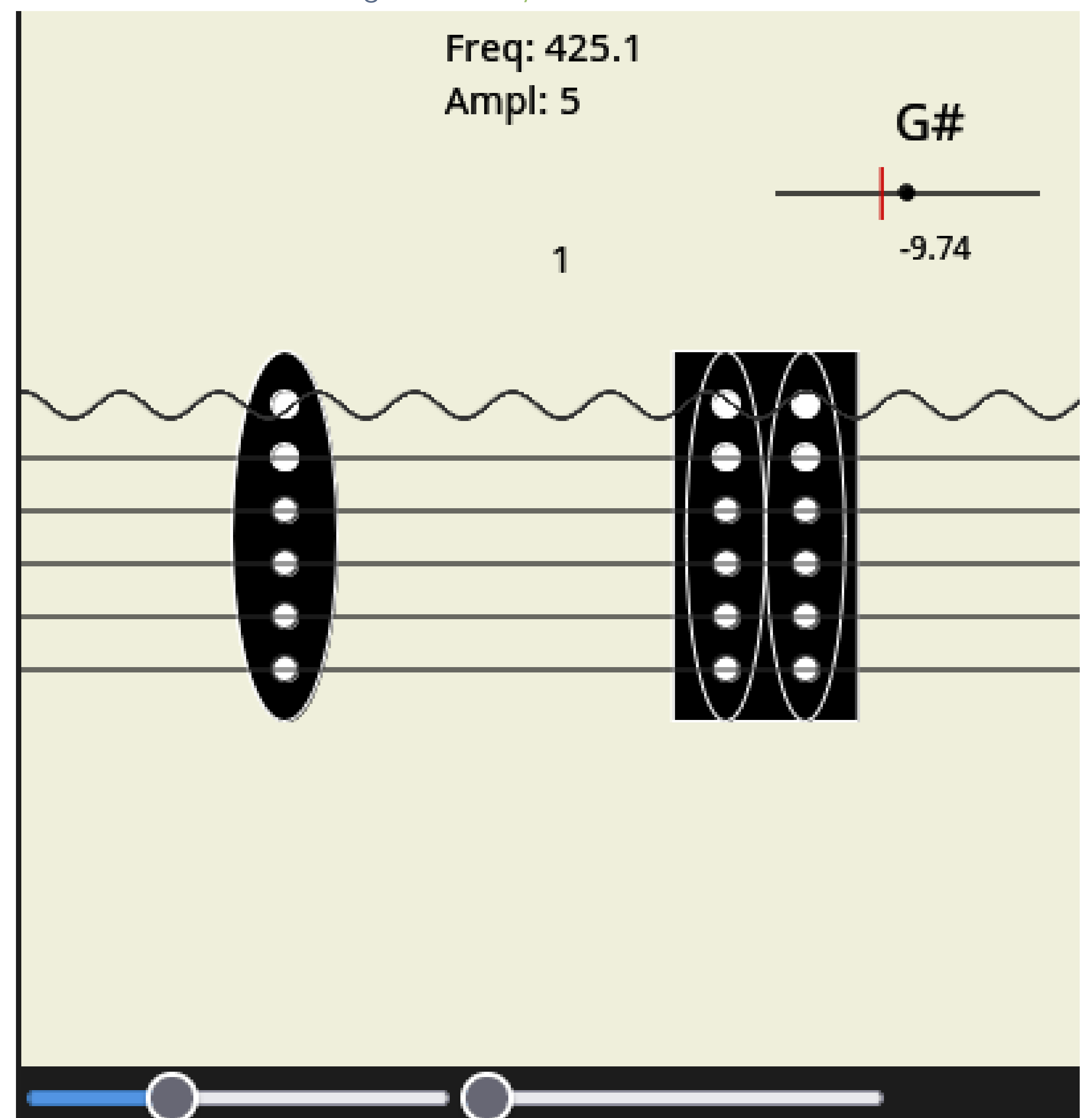
Ou seja, a frequência base da nota multiplicada por 2 elevado ao número da oitava menos um. Basicamente, o afinador funciona com esse cálculo para determinar a oitava da nota, a nota mais próxima e a distância até ela. Isso permite que o traço vermelho do afinador indique a distância, positiva ou negativa, conforme a direção.

Para o som, usei a biblioteca **Oscillator** do P5, que gera um som com a frequência e amplitude fornecidas. Além disso, implementei uma transição para que a onda, visualmente, perca amplitude ao longo do tempo, acompanhando a diminuição do som.

## Resultado

Essa é uma ideia de como o meu projeto ficou, com essa imagem pode-se observar que a primeira corda está selecionada e criando uma onda com amplitude 5 e frequência de aproximadamente 425hz, que de acordo com o nosso afinador à direita, a nota que ela representa é um **Sol Sustenido** que está a uma distancia de aproximadamente 10hz da nota exata.

Figura: Simulação em andamento



## Tecnologia

Todo o trabalho foi feito utilizando a própria IDE online do P5.js, que em base é uma portabilidade do Processing para a web utilizando como linguagem base o javascript, além das funções básicas do próprio P5.js, também utilizo o **Oscillator** que é uma classe dentro do P5.

## Referências

- [Muted.io](https://muted.io)
- [Liutaiomottola.com](https://liutaiomottola.com)
- [Electricity-magnetism.org](https://electricity-magnetism.org)