

# UNIVERSIDAD DE GUANAJUATO



**Universidad de Guanajuato  
Campus Irapuato-Salamanca**

**DICIS**

**Licenciatura en Artes Digitales**

**Edición de Audio Digital**

**Proyecto Final:**

**Max 8 Patcher: "Mezcladora" (Mixer)**

**Original: May 25th 2020**

**Translation: June 15th 2020**

**Equipo #**

**González Uribe Hugo Alberto**

**Espadas Chacón Carlos Daniel**

**Jacobo Flores Filemón**

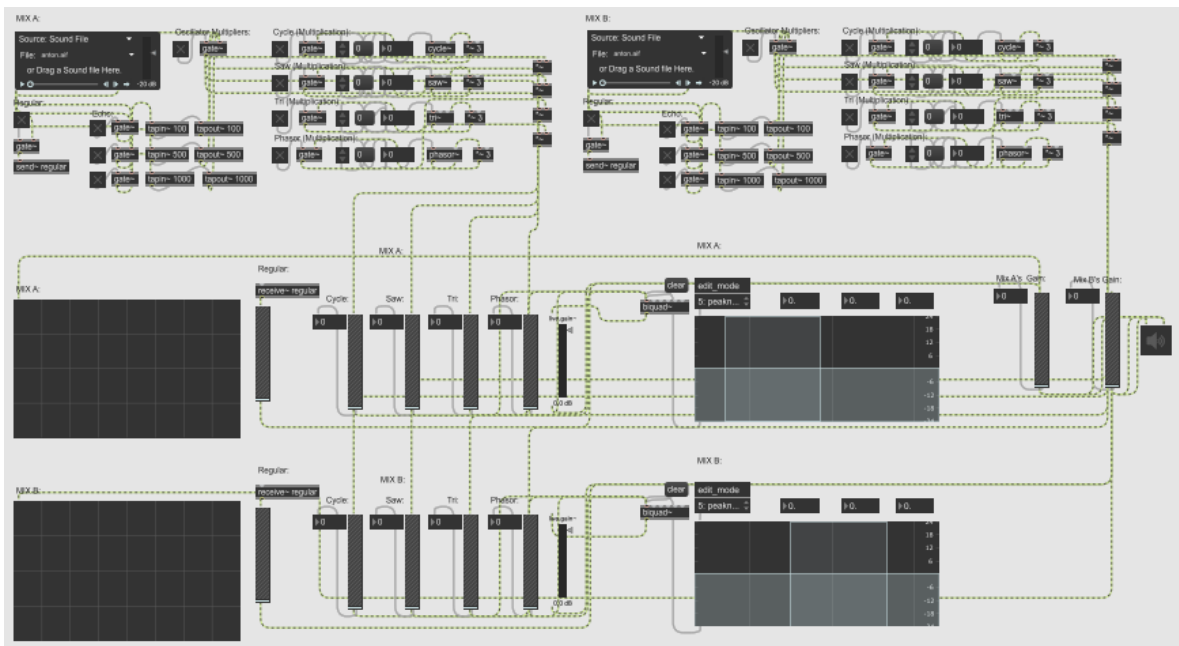
**Llamas Rangel Karla Fernanda**

## Index

|   |   |
|---|---|
| Description:  | 2 |
| Functionality:  | 2 |
| 1. Part 1: File Selection and Regular Play.           | 3 |
| 2. Part 2: Echo.                                      | 3 |
| 3. Part 3: Oscillators.                               | 4 |
| 4. Part 4: Oscillator Gain.                           | 5 |
| 5. Part 5: Equalizers.                                | 5 |
| 6. Part 6: Mixes, Signal Visualization and Final Mix. | 5 |
| 7. Part 7: Workflow Maps.                             | 6 |

GITHUB link: [https://github.com/insert-edgelordname/HAGU-LEARN/tree/master/MAX8\\_Mixer\\_Patcher](https://github.com/insert-edgelordname/HAGU-LEARN/tree/master/MAX8_Mixer_Patcher)

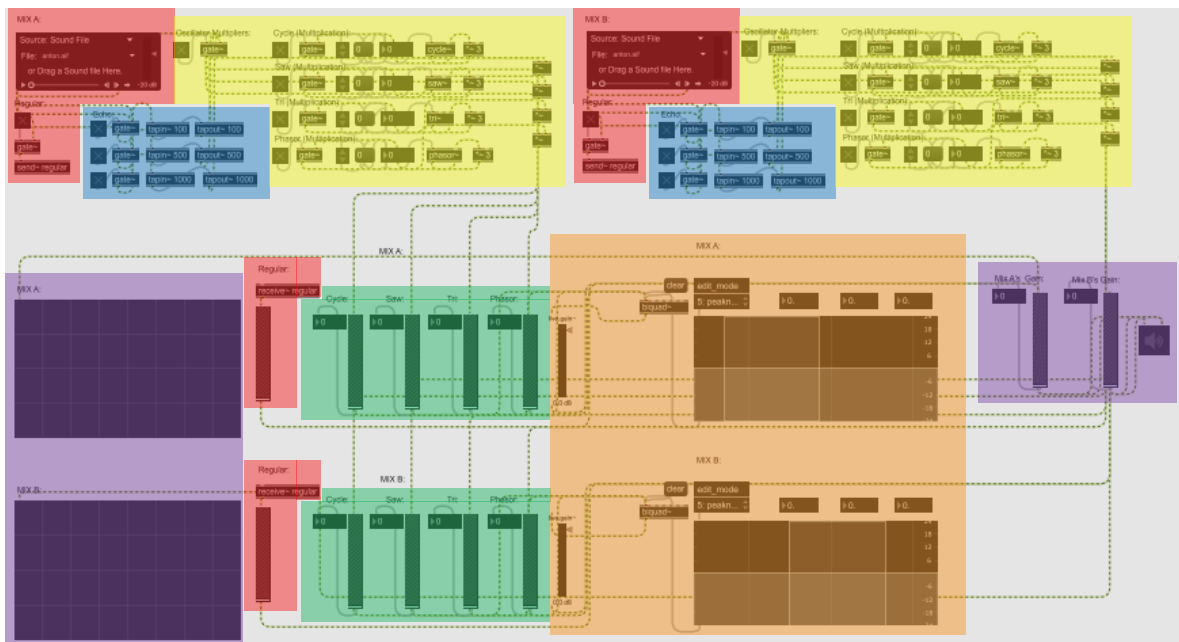
## Description:



The created patcher is a mixer in real time, giving the user the possibility of modifying the amplitude of two audio files, allowing to even change the proportion and degree in which each wave is affected, while also giving the chance to implement echo effects and signal modification through the use of equalizers, producing a new experience.

## Functionality:

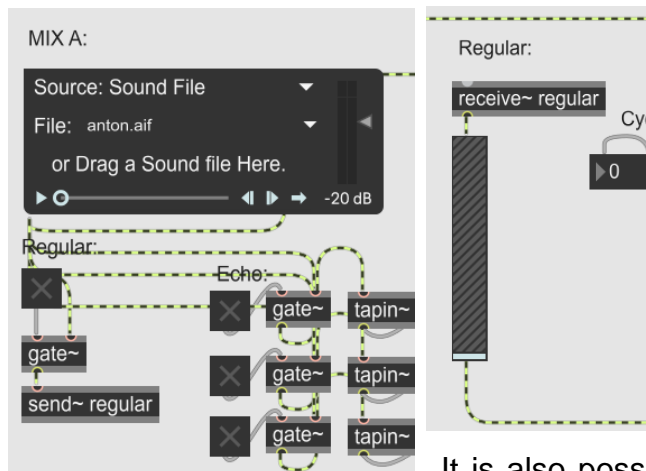
For the purpose of simplicity within the explanation, we will divide the patcher into 6 segments, and each one Will be explained in detail, be aware that within the patcher



lots of functions and processes are repeated, specially between the two channels (Mix A and Mix B), being these two practically identical, so we will unify the explanation, due to the fact that explaining each component of each channel would end up resulting redundant.

1. **Part 1: File Selection and Regular Play.**
2. **Part 2: Echo.**
3. **Part 3: Oscillators.**
4. **Part 4: Oscillator Gain.**
5. **Part 5: Equalizers.**
6. **Part 6: Mixes, Signal Visualization and Final Mix.**
7. **Part 7: Workflow Maps (the patcher's workflow).**

### Part 1: File Selection and Regular Play.

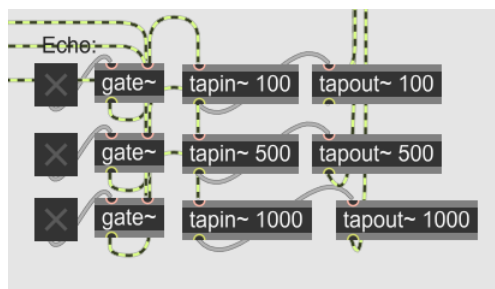


Here is where the mixer *patcher* begins, it makes use of a *bpatcher* to allow the user to load in any audio file.

The Mixer patcher can reproduce audio normally, without alterations to it; this is accomplished through a toggle button and a *gate~* function which allow for the audio to pass normally if said toggle is triggered.

It is also possible to alter the gain of this unaltered signal by using the slider shown.

### Part 2: Echo.

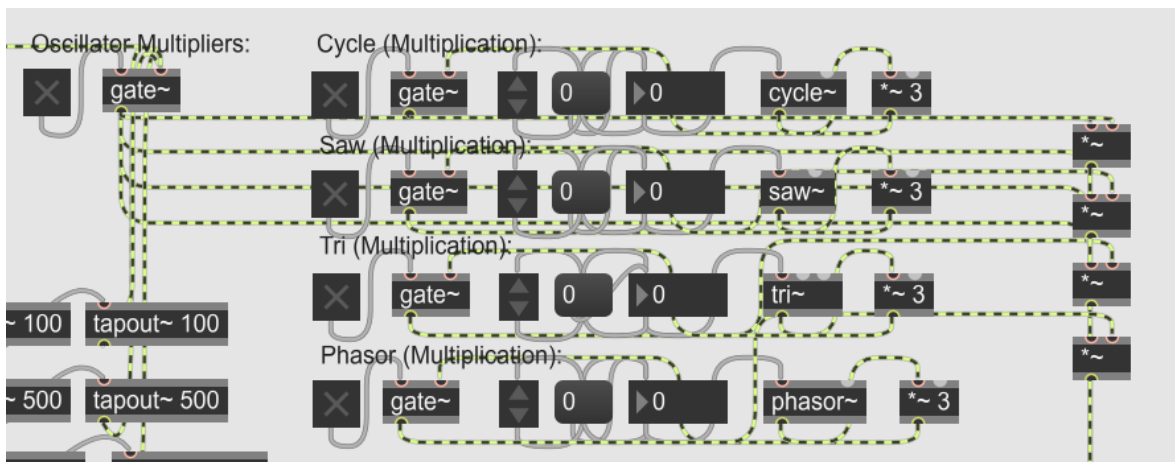


Immediately after the regular playing process, we have the chance of implementing echoes, with three options: echo with a 100-millisecond delay, 500-millisecond delay, and a 1000-millisecond delay.

The activation process for the echoes is the same as the one for unaltered reproduction, the respective toggle, which will allow the signal to pass to the corresponding gate, and receive the alteration indicated. The echoes are not mutually exclusive, so it is possible to turn them all on at once, if so is the user's will.

If one of the toggles has been turned on, said signal will be sent from the *tapin~*, to the *tapout~* to the next section in the Mixer, **the oscillators**, to allow the user to apply them, if they so choose, to the echo.

### Parte 3: Osciladores.



This is the most complex part of the whole program, so it will be divided into various sections.

#### 1. Oscillator Activation:

Just like previous parts, they are enabled through the toggles and gates~, first, the user must activate the main toggle (found just below the comment “Oscillator Multipliers”), and soon after, the desired oscillator should be turned on, through its respective toggle.

There are 4 oscillator types available:

- Sine (Cycle).
- Sawtooth (Saw).
- Tri.
- Phasor.

#### 2. Frequency Variations:

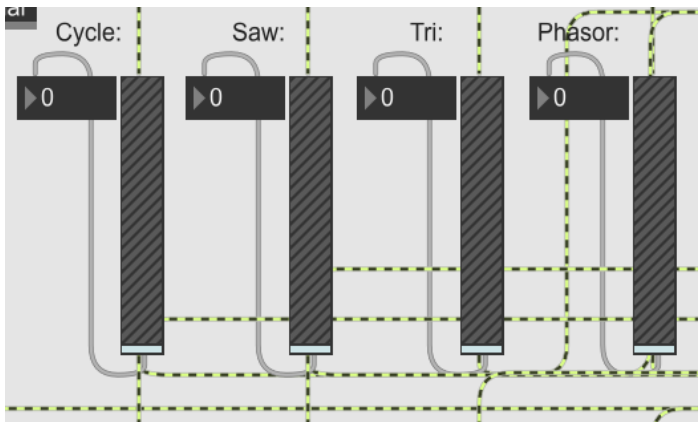
It is possible to alter the frequency of each one of the oscillators, be it in a precise manner, by using the arrow buttons that will increment or decrement the value in the numeric, or interacting with said numeric directly; it is even possible to reset the value to “0” by clicking on the message “0”, right next to the numeric.

If the user only wants to apply the echo to the signal, it is suggested to apply the “cycle” oscillator, and leave its frequency as zero, causing the signal to come out unaltered, only applying the echo.

#### 3. Applying Oscillators to the Waves:

Once determined the frequency values, the resulting wave is multiplied by three, it is then multiplied by the original audio file, and the next section starts operating, **the oscillator gain.**

#### Part 4: Oscillator Gain.

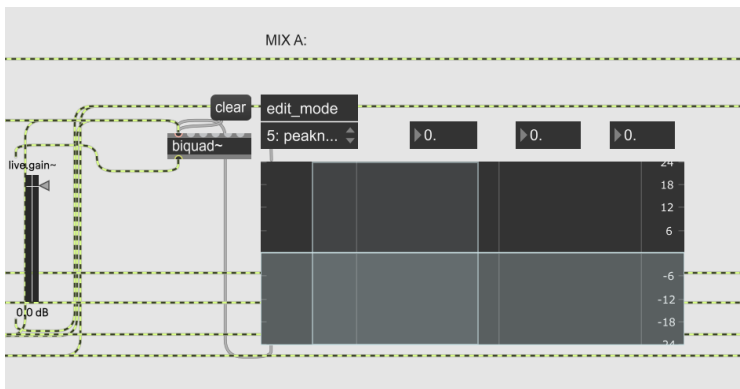


Once the desired oscillators are setup, it is possible to change the gain of said oscillators, individually, in real time, allowing for control over the “presence” of each oscillator in the mix.

Once the gain sliders are properly setup, the signal is sent to the next segment of the

program, **the equalizers.**

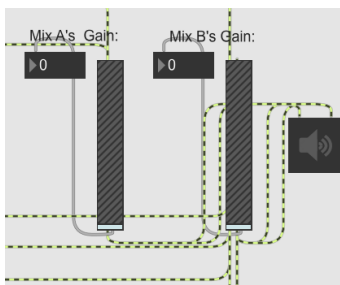
#### Part 5: Equalizers.



Each channel or “Mix”, has an equalizer that can take whatever shape the user desires, be it a highpasser or a basspasser, there is complete control over the produced audio.

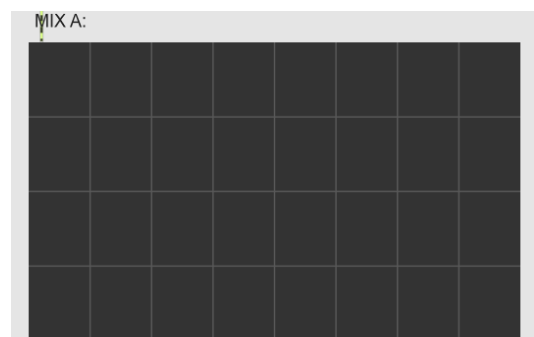
Alongside each equalizer, there is a live gain slider, which shows the levels that the signal reaches, this live gain slider can also be altered to increase or decrease the signal levels. The produced signal is sent, lastly, to the corresponding **mix** gain slider.

#### Part 6: Mixes, Wave Visualization and Final Mix.



Finally, we have the gain sliders for both signals, “Mix A” and “Mix B”, which allow the user to alter the gain of each individual Mix or Channel. To conclude, as the final patcher element, the wave

visualization; the graphics, one per mix, are found at the far left, and are connected to their respective final Mix Gain sliders, thus, showing each wave as it is being played, in real time.



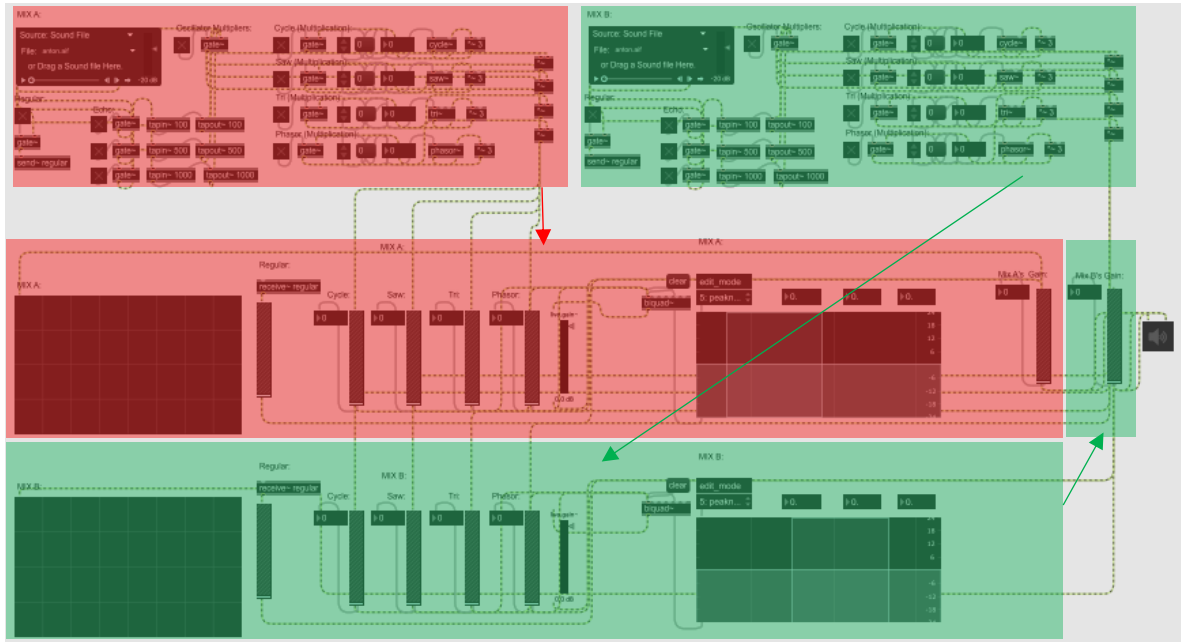
## Part 7: Workflow Maps.

Due to the sheer amount of elements at work, and the placement of them in relation to one another, this document features a map showing the sections that belong to each mix, alongside two maps showing the workflow of each mix/channel within the patcher.

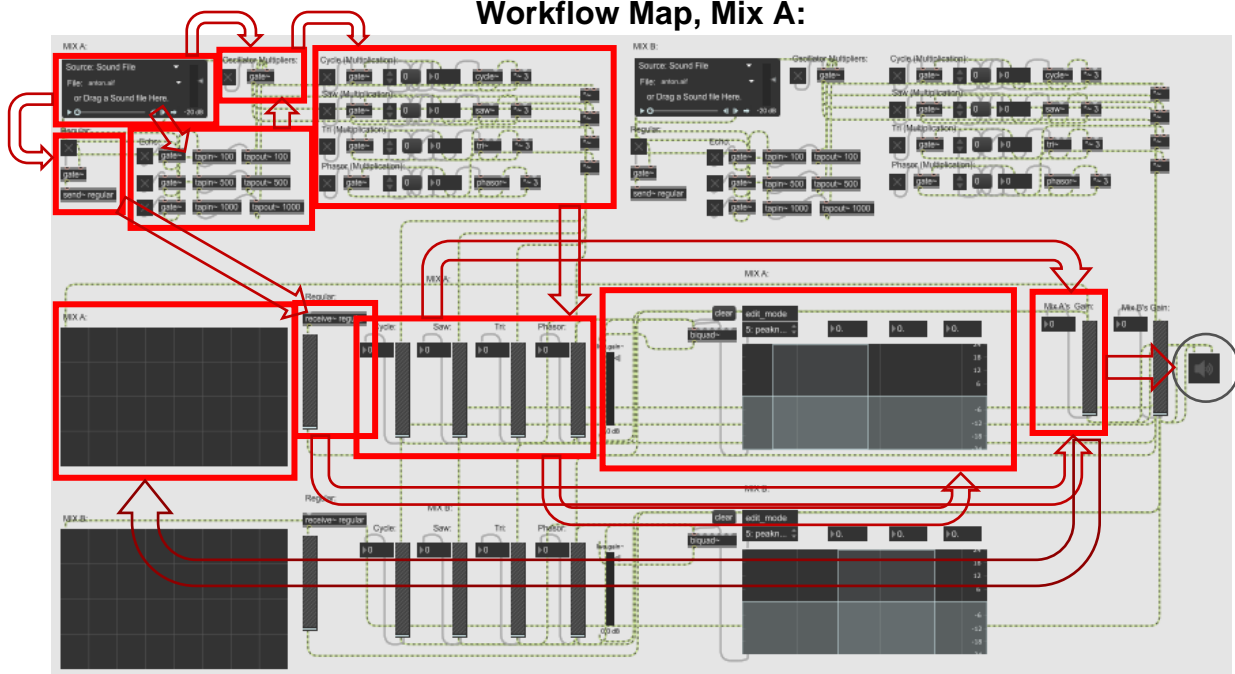
**Mixes Map:**

**Red = Mix A**

**Green = Mix B**



**Workflow Map, Mix A:**



## Workflow, Mix B:

