Contenido del tutorial:

1. **Objetos y Operaciones Básicas:**
   Introducción a los objetos en Pure Data. Aprender a sumar, multiplicar, y manipular números utilizando `float`, `+`, `*`, y `message`.

2. **Oscilador Básico y Control de Frecuencia y Volumen:**
   Uso del objeto `osc~` para generar ondas sinusoidales, control de frecuencia con `number` y ajuste de volumen con `*~` y `dac~`.

3. **Arrays y Representación de Ondas:**
   Crear y visualizar una onda en un array usando `bang`, `metro`, y `trigger`, aplicando una representación gráfica de la señal de audio.

4. **Tipos de Ondas (Noise y Otras):**
   Experimentar con diferentes tipos de ondas como `noise~`, `phasor~`, `triangle~`, y comparar sus características.

5. **Modulación de Frecuencia (FM) y Amplitud (AM):**
   Aplicar modulación de frecuencia (FM) y amplitud (AM) utilizando un oscilador modulador que modifique la señal de otro oscilador principal.

6. **Envolventes ADSR con `line`:**
   Crear una envolvente ADSR para controlar la evolución de un sonido mediante el objeto `line` y aplicar dicha envolvente sobre la amplitud del oscilador.

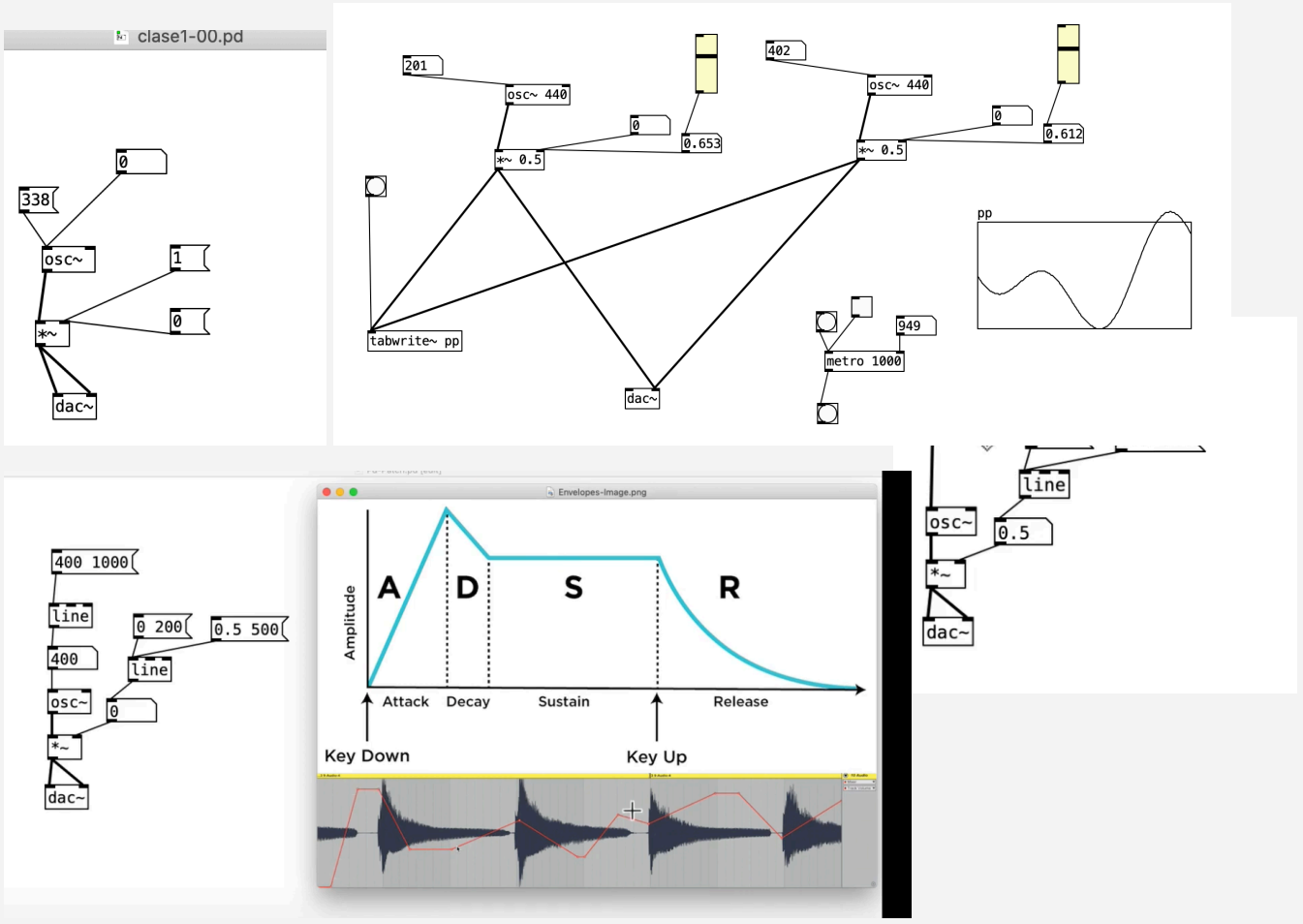
7. **Síntesis Aditiva:**
   Generar un sonido complejo sumando varias ondas sinusoidales con diferentes frecuencias y amplitudes, usando `osc~` y `+~`.

8. **Síntesis Sustractiva:**
   Partir de una señal rica en armónicos (como `noise~` o `saw~`) y aplicar un filtro (`lop~`) para eliminar frecuencias y obtener el sonido deseado.

9. **Modulación con LFO (Low Frequency Oscillator):**
   Usar un oscilador de baja frecuencia (LFO) para modular parámetros de otro oscilador, como la frecuencia o la amplitud, creando efectos de vibrato o trémolo.

10. **Proyecto Final - Síntesis Completa:**

    Integrar todos los elementos aprendidos para crear un sintetizador completo. Los estudiantes usarán osciladores, filtros, modulación, envolventes y control en tiempo real para construir un instrumento que generará sonidos complejos.



1. Introducción a los Objetos y Operaciones Básicas

Explicación:

- En **Pure Data**, los **objetos** son los bloques fundamentales que realizan acciones. Los objetos se crean escribiendo su nombre en una caja.
- **Objetos básicos:** `+`, `-`, `*`, `/`, que permiten operaciones matemáticas sencillas.
- **Mensajes:** Envían información a otros objetos. Se crean con cajas de mensaje (`message`).
- **Números y floats:** Controlan valores numéricos. `float` permite decimales.

Ejercicio 1:

1. Crea dos objetos de tipo `float`, uno para cada número.
2. Crea un objeto `+` y conéctalo a los dos floats.
3. Muestra el resultado con otro objeto de número (`number`).

- **Extensión:** Agrega más objetos para multiplicar (`*`) y dividir (`/`) y observa los resultados.

---

2. Oscilador Básico: Control de Frecuencia y Volumen

Explicación:

- El objeto `osc~` genera una onda sinusoidal con una frecuencia determinada.
- Para controlar el volumen, utilizamos `*~` para multiplicar la señal de audio por un valor entre 0 y 1 (donde 0 es silencio y 1 es el volumen máximo).
- El objeto `dac~` envía el sonido a los altavoces.

Ejercicio 2:

1. Crea un `osc~` para generar una onda sinusoidal.
2. Conéctalo a `*~` para controlar el volumen.
3. Luego conéctalo a `dac~` para que el sonido se reproduzca en los altavoces.
4. Usa un `number` para ajustar la frecuencia del `osc~` y otro para ajustar el volumen en `*~`.

---

3. Arrays y Representación de Onda

Explicación:

- Un **array** permite almacenar y visualizar datos, como una onda.
- Usa `bang` para ejecutar algo, `metro` para crear un pulso repetitivo y `trigger` para controlar el flujo de mensajes.

Ejercicio 3:

1. Crea un array para mostrar la forma de onda de un `osc~`.
2. Usa un `metro` y `bang` para disparar un `trigger` que actualice la visualización de la onda.

---

4. Tipos de Ondas: Noise y Otros

Explicación:

- Pure Data ofrece varios tipos de ondas, como `noise~` que genera ruido blanco.
- Experimenta con otras formas de onda como `phasor~`, `triangle~`, etc.

Ejercicio 4:

1. Crea un objeto `noise~` y conéctalo a `dac~`.
2. Cambia la onda a `phasor~` o `triangle~` y compara el sonido.

---

5. Modulación de Frecuencia (FM) y Amplitud (AM)

Explicación:

- **Modulación de Frecuencia (FM):** Se usa un oscilador para modular la frecuencia de otro.
- **Modulación de Amplitud (AM):** Se usa un oscilador para modular la amplitud (volumen) de otro.

Ejercicio 5:

1. Crea dos osciladores `osc~`, uno para la señal portadora (principal) y otro para la señal moduladora (modulador).

2. Conéctalos para que uno module la frecuencia del otro (FM) y luego la amplitud (AM).

---

6. ADSR con `line`

Explicación:

- **ADSR** (Attack, Decay, Sustain, Release) es un tipo de envolvente usada para controlar la evolución de un sonido.
- En Pure Data, `line` puede generar una envolvente simple.

Ejercicio 6:

1. Crea un `osc~` para la onda principal.
2. Usa `line` para crear una envolvente que controle la amplitud del `osc~`.
3. Conecta la salida de `line` a `*~` para aplicar la envolvente al volumen.

- **Extensión:** Ajusta los valores de ataque, decaimiento, sostenido y liberación para observar cómo afecta al sonido.

7. Síntesis Aditiva

Explicación:

- **Síntesis Aditiva:** Se genera un sonido sumando varias ondas sinusoidales con diferentes frecuencias y amplitudes.
- En este ejercicio, los estudiantes combinarán varios `osc~` para crear un sonido complejo.

Ejercicio 7:

1. Crea tres osciladores `osc~` con diferentes frecuencias (por ejemplo, 200 Hz, 400 Hz, y 600 Hz).
2. Ajusta las amplitudes de cada oscilador usando `*~` y suma las señales utilizando `+~`.
3. Conecta el resultado a `dac~` para escuchar el sonido aditivo.

- **Extensión:** Experimenta con diferentes frecuencias y amplitudes para crear diferentes timbres.

---

8. Síntesis Sustractiva

Explicación:

- **Síntesis Sustractiva:** Se parte de una señal rica en armónicos (como `noise~` o `saw~`) y se filtra para eliminar ciertas frecuencias, obteniendo el sonido deseado.
- Usaremos filtros (`lop~`, `hip~`) para modificar la señal.

Ejercicio 8:

1. Crea una señal rica en armónicos usando `noise~` o `saw~`.
2. Usa un filtro `lop~` (filtro pasa bajos) para eliminar frecuencias altas.
3. Conecta la señal filtrada a `dac~`.

- **Extensión:** Cambia el corte del filtro (`lop~`) en tiempo real para explorar cómo afecta el sonido.

---

9. LFO (Low Frequency Oscillator)

Explicación:

- **LFO:** Un oscilador de baja frecuencia modula otro parámetro, como la frecuencia o amplitud de un oscilador principal, creando efectos como vibrato o trémolo.
- En este ejercicio, los estudiantes crearán un LFO que modula la frecuencia de un oscilador.

Ejercicio 9:

1. Crea un `osc~` de baja frecuencia (por ejemplo, 0.5 Hz) y úsalo para modular la frecuencia de otro `osc~` (por ejemplo, 440 Hz).
2. Conecta el oscilador modulado a `dac~` para escuchar el efecto de vibrato.

- **Extensión:** Experimenta con diferentes frecuencias del LFO y observa cómo afectan el sonido.

---

10. Proyecto Final: Creación de un Síntesis Completa

Explicación:

- En este proyecto final, los estudiantes combinarán todo lo que han aprendido para crear un sintetizador completo que incluya modulación, envolventes, filtros y tipos de síntesis.
- Se espera que los estudiantes completen este proyecto en casa y lo presenten en la siguiente clase.

Instrucciones:

1. **Osciladores:** Crea un sintetizador con al menos dos osciladores (`osc~`) que generen ondas sinusoidales o de otro tipo (`noise~`, `phasor~`).
2. **Síntesis Aditiva y Sustractiva:** Suma o filtra las señales para crear timbres complejos utilizando los principios de síntesis aditiva y sustractiva.
3. **Modulación:** Implementa modulación FM o AM, utilizando al menos un LFO para variar la frecuencia o amplitud de uno de los osciladores.
4. **Envolvente ADSR:** Crea una envolvente ADSR utilizando `line` o `vline~` para controlar la amplitud de la señal final.
5. **Control en Tiempo Real:** Usa controles (sliders, knobs) para ajustar parámetros como frecuencia, amplitud, corte del filtro, y velocidad del LFO en tiempo real.

Requisitos del Proyecto Final:

- **Mínimo:** Dos osciladores, un filtro, un LFO y una envolvente ADSR.
- **Opcional:** Agregar efectos adicionales como reverberación (`freeverb~`), delay, o distorsión.
- **Presentación:** Los estudiantes deben estar preparados para explicar su patch y cómo cada componente contribuye al sonido final.