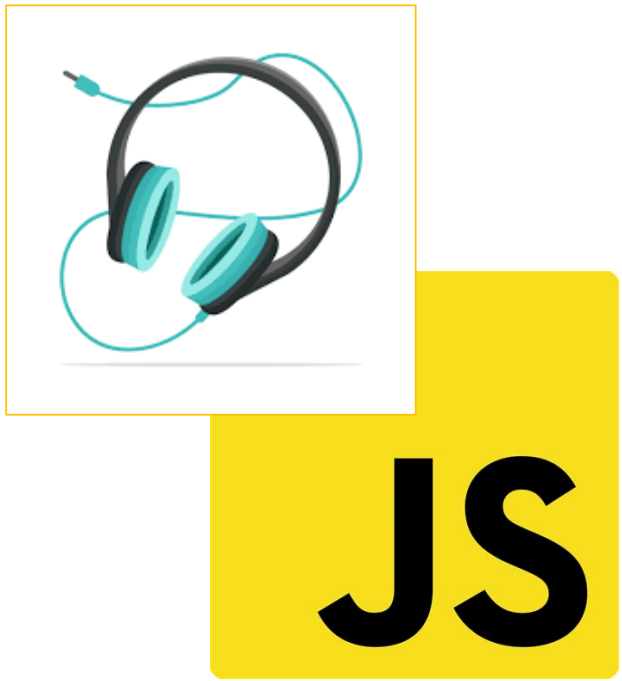
IPO

Interacción Persona Ordenador

E.T.S. Ingeniería Informática
Universidad de Sevilla

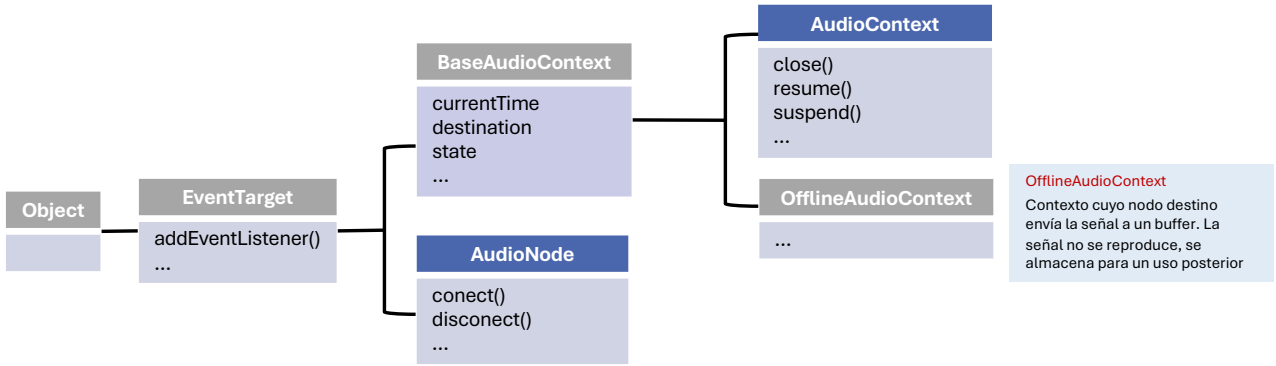
Web Audio API



1. Contexto de audio
2. Osciladores
3. Parámetros de audio
4. Nodos intermedios

Contexto de audio: Introducción

Web Audio API: Conjunto de interfaces especializadas en la gestión del sonido en la web.



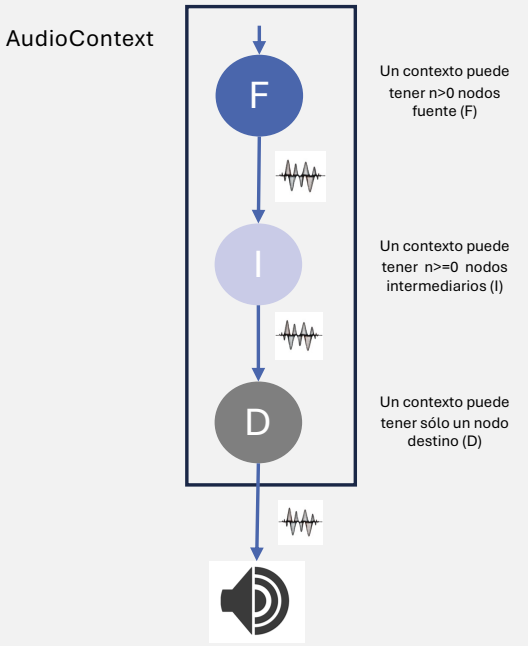
Contexto de audio: Objeto que almacena y gestiona un grafo de nodos de audio

AudioContext
- Contexto pensado para la reproducción de audio en tiempo real
- El nodo destino envía la señal a los altavoces/micrófonos del sistema
- Permite detener/reanudar/terminar la reproducción

AudioNode
- Interfaz para la gestión de nodos de audio
- Ofrece métodos para (des)conectar el nodo en un contexto de audio

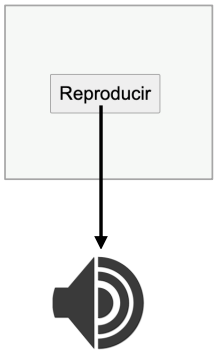
BaseAudioContext interfaz común para los contextos de audio
- currentTime (*double*) temporizador en segundos
- destination referencia al audio nodo de destino del contexto
- state (closed, interrupted, running, suspended)

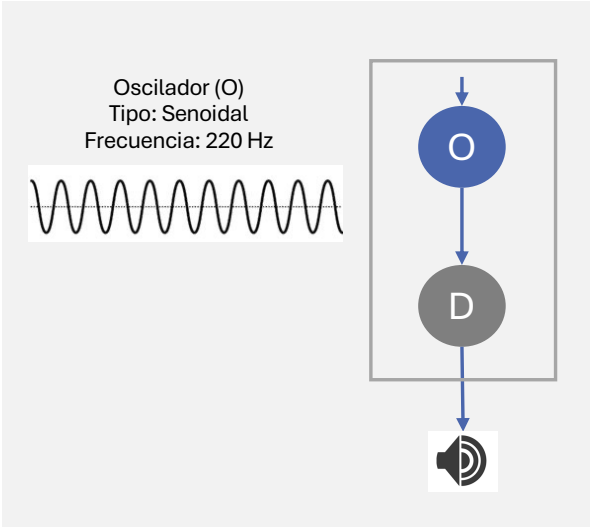
La señal de audio procede de los **nodos fuentes** y puede ser manipulada a través de varios **nodos intermedios** (efectos o filtros, analizadores, etc.) hasta alcanzar el **nodo de destino**



Contexto de audio: Ejemplo

```html
<body>
    <button class="btnReproducir" id="btnReproducir">Reproducir</button>
</body>
```






```js
const audioCtx = new AudioContext();

const btnReproducir = document.getElementById("btnReproducir");
btnReproducir.addEventListener("click", reproducirAudio);

función reproducirAudio() {

        const source = new OscillatorNode(audioCtx, {
                type: "sine",
                frequency: 220,
        });

        source.connect(audioCtx.destination);

        const inicio = audioCtx.currentTime;
        const duracion = 4;
        source.start(inicio);
        source.stop(inicio + duracion);
}
```

Creación del nodo fuente

Conexión al nodo destino

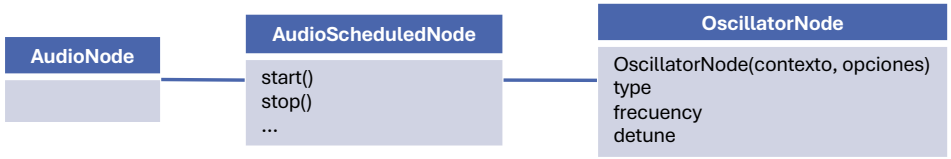
Reproducción de la señal

Los métodos start() y stop() permiten comenzar y finalizar la reproducción de audio

Para evitar una experiencia molesta, el sonido está por defecto inhabilitado (estado suspended)
El usuario debe dar permiso para comenzar la reproducción. Ejemplo: clic en un botón

AudioContext.currentTime
- La primera vez se crea el temporizador con valor inicial 0
- Las siguientes veces devuelve el tiempo transcurrido en segundos desde la creación del temporizador

Osciladores



OscillatorNode (Osciladores) Nodos de audio fuente que generan una señal periódica

AudioScheduledNode interfaz común para los nodos de audio fuente que admiten fijar el tiempo inicial start(tiempo) y final stop(tiempo) de la reproducción

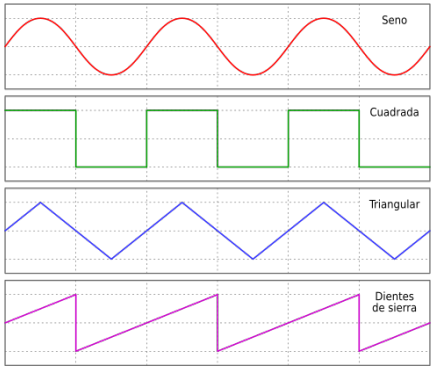
Una vez parada la reproducción, no puede reanudarse

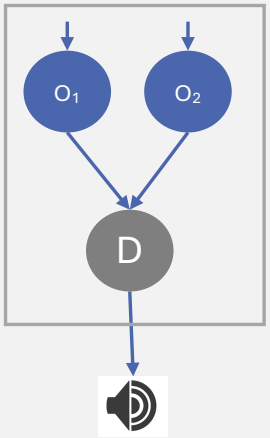
type Forma de la onda: sine, square, sawtooth, triangle,
frequency Frecuencia en Hercios (Hz)
detune Reajuste de la frecuencia en *cents* (centésimas de semitono)

Valores por defecto:
señal sinusoidal a 440 Hz (La4)

Señal (Formas de onda)

Hz: Número de ciclos por segundo






```
const audioCtx = new window.AudioContext();

const btnReproducir = document.getElementById("btnReproducir");
btnReproducir.addEventListener("click", reproducirAudio);

function reproducirAudio() {

  const oscillator01 = new OscillatorNode(audioCtx, {
    type: "sine",
    frequency: 220,
  });

  const oscillator02 = new OscillatorNode(audioCtx, {
    type: "triangle",
    frequency: 210,
  });
;
  oscillator01.connect(audioCtx.destination);
  oscillator02.connect(audioCtx.destination);

  const now = audioCtx.currentTime;

  const inicioOscillator01 = now;
  const inicioOscillator02 = now + 1;
  const duracionOscillator01 = 4;
  const duracionOscillator02 = 3;


  oscillator01.start(inicioOscillator01);
  oscillator01.stop(inicioOscillator01 + duracionOscillator01);

  oscillator02.start(inicioOscillator02);
  oscillator02.stop(inicioOscillator02 + duracionOscillator02);
}
```

Parámetros de audio

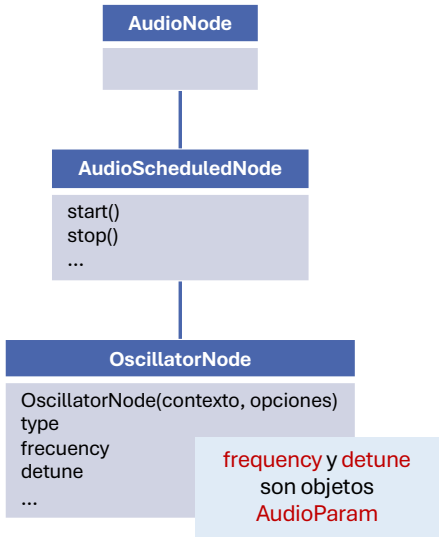
AudioParam

setValueAtTime()
linearRampToValueAtTime()
exponentialRampToValueAtTime()
...

AudioParam: interfaz para controlar y automatizar los parámetros de los nodos de audio

- Cambios instantáneos: setValueAtTime(value, time)
- Cambios progresivos (lineal y exponencial):
  - linearRampToValueAtTime(value, endTime)
  - exponentialRampToValueAtTime(value, endTime)

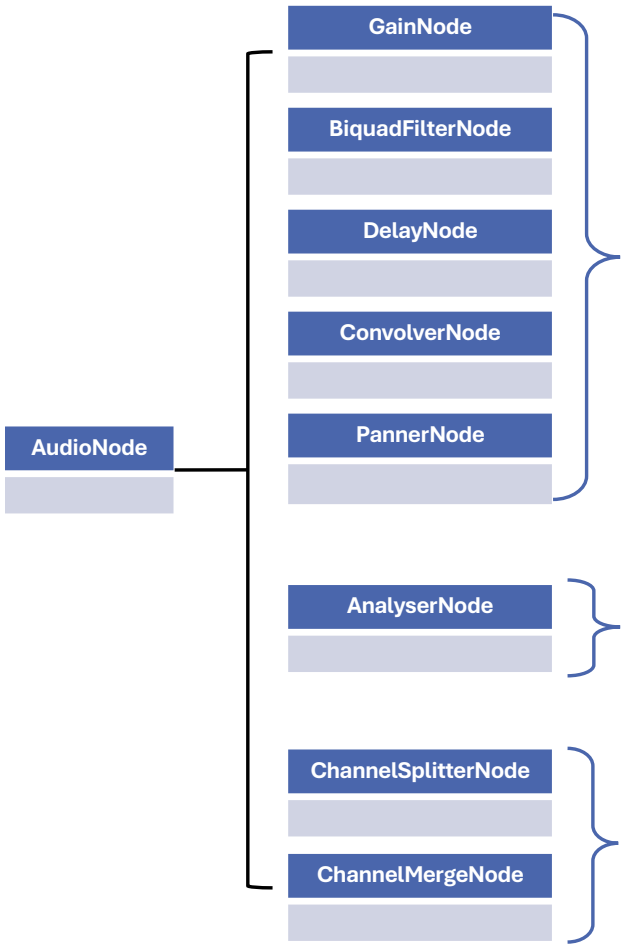
- Los valores temporales son agendados para realizar los cambios en orden cronológico (no influye el orden de aparición en el código)
- Los cambios progresivos fijan el instante final y de inician en el instante agendado inmediatamente anterior




```
// Frecuencias del acorde do Mayor: do4, mi4, sol4, do5
const do4 = 261.63;
const mi4 = 329.63;
const sol4 = 392.00;
const do5 = 523.25;

const duracionNota = 1.5;

let tiempo = audioCtx.currentTime;
source.frequency.setValueAtTime(do4, tiempo);

tiempo += duracionNota;
source.frequency.setValueAtTime(mi4, tiempo);

tiempo += duracionNota;
source.frequency.setValueAtTime(sol4, tiempo);

tiempo += duracionNota;
source.frequency.setValueAtTime(do5, tiempo);

source.start();
tiempo += duracionNota;
source.stop(tiempo);
```

```
// Frecuencias del acorde do Mayor: do4, mi4, sol4
const do4 = 261.63;
const mi4 = 329.63;
const sol4 = 392.00;

const duracionExponential = 1.5;
const duracionEstatica = 2;
const duracionLinear = 1.5;

let tiempo = audioCtx.currentTime;
source.frequency.setValueAtTime(do4, tiempo);

// Fija tiempo de inicio para el cambio exponecial
tiempo += duracionEstatica;
source.frequency.setValueAtTime(do4, tiempo);

tiempo += duracionExponential;
source.frequency.exponentialRampToValueAtTime(mi4, tiempo);

  // Fija tiempo de inicio para el cambio lineal
tiempo += duracionEstatica;
source.frequency.setValueAtTime(mi4, tiempo);

tiempo += duracionLinear;
source.frequency.linearRampToValueAtTime(sol4, tiempo);

source.start();
tiempo += duracionEstatica;
source.stop(tiempo);
```

Nodos intermedios: Interfaces



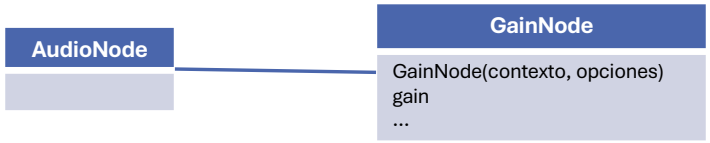
Nodos modificadores: Modifican las propiedades del sonido:

- Volumen
- Frecuencia (ecualización o efectos tonales)
- Tiempo de retraso (efecto de eco)
- Reverberación
- Posición espacial (panorámica)

Nodo analizador: Proporcionan datos en tiempo real sobre la señal de audio (sin modificarla). Se emplea en visualización del sonido.

Nodos de enrutamiento/mezcla: Permiten separar/combinar canales de audio para procesamiento avanzado del sonido

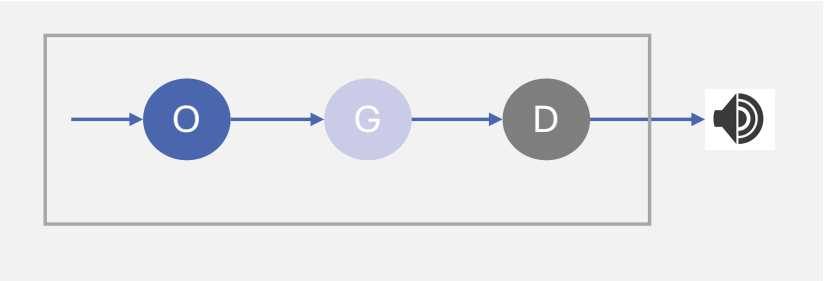
Nodos intermedios: GainNode



GainNode: Controla el **volumen** (ganancia) de la señal de audio que pasa a través de él

gain Factor de multiplicación aplicado a la señal de audio.

- Un valor de 1.0 (por defecto) significa que la señal pasa sin cambios
- Valores mayores a 1.0 amplifican el sonido
- Valores entre 0.0 y 1.0 lo atenúan




```
const oscilador = new OscillatorNode(audioCtx, {
  type: "sine",
  frequency: 300,
});

const ganancia = new GainNode(audioCtx, {
  gain: 0.2,
});

oscilador.connect(ganancia);
ganancia.connect(audioCtx.destination);

const inicio = audioCtx.currentTime;
const tiempoGanancia = inicio + 2;
const tiempoFinal = tiempoGanancia + 2;

ganancia.gain.setValueAtTime(0.9, tiempoGanancia);

oscilador.start(inicio);
oscilador.stop(tiempoFinal);
```

Primero se fija un valor muy por debajo de uno para subirlo posteriormente, pero sin llegar a uno para evitar que el sonido se sobresature.


UNIVERSIDAD DE SEVILLA
UNIVERSIDAD Ð SEVILLA