

clase-02

repaso clase anterior y programa hoy (5 min)

la clase pasada aprendimos:

- presentaciones
- contexto artes mediales y Arduino
- Git, GitHub y Markdown

hoy aprenderemos:

- señales analógicas y digitales
- computadores y microcontroladores
- programar semáforo usando Processing

señales analógicas y digitales (45 min)

supondremos que nuestras señales son del mundo real:

- señales unidimensionales
- señales en función del tiempo t

más (demasiada) información:

- señales y sistemas, Alan V. Oppenheim y Alan S. Willsky.

señales analógicas

las señales analógicas tienen **valores continuos** en **todo momento**:

- notación $y(t)$
- t es continuo
- $y(t)$ es continuo

ejemplos de señales analógicas:

- presión atmosférica
- fotografía polaroid
- temperatura de un lugar
- sonido en un disco de vinilo
- salida de un micrófono

análogo significa similar, porque las señales analógicas también se parecen a otras.

una señal analógica se puede obtener desde un sensor,

ejemplos de transductores / sensores análogos:

| sensor | entrada | salida |
|--------|---------|--------|
|--------|---------|--------|

| sensor | entrada | salida |
|---------------------|---------------------|-------------|
| micrófono | presión atmosférica | voltaje |
| cápsula de guitarra | vibración de cuerda | voltaje |
| fotoresistor | intensidad de luz | resistencia |
| perilla | posición (ángulo) | resistencia |

la transducción no es perfecta, siempre introduce errores y distorsión.

esto no necesariamente es malo, de hecho en música estas diferencias resultan en sabores musicales distintos. más info sobre distintos tipos de compresores <https://reverb.com/news/what-are-the-types-of-compressor-effects-the-basics>

señales digitales

las señales digitales tienen **valores discretos** en **momentos discretos**:

- notación $y[n]$
- n es discreto
- $y(t)$ es discreto

ejemplos de señales digitales:

- sonido en un disco compacto
- imagen en un computador
- tiempo en un reloj digital

conjuntos y sistemas de números (30 min)

conjuntos:

- números naturales: los que contamos con los dedos. (supuesto: cada unidad es equivalente) por qué contamos con diez dedos? 1,2,3,...
- números enteros: números que pueden ser escritos sin fracciones ..., -3, -2, -1, 0, 1, 2, 3, ...
- números racionales: números que pueden ser escritos como fracciones de dos números enteros.
- números reales: números con parte decimal, sirven para medir distancias.
- números irracionales: números que son reales, pero no racionales, como $\sqrt{2}$ y π .
- números complejos: números que poseen coordenadas reales e imaginarias.

demonstración interesante:

- los números naturales son infinitos.
- los números enteros se pueden contar con naturales, son igualmente infinitos.
- los números racionales son aún más infinitos.

sistemas:

- decimal: base 10
- binario: base 2

- hexadecimal: base 16

actividad: escribamos los primeros 20 números, empezando desde 0, usando los sistemas decimal, binario y hexadecimal.

pausa: materiales (10 min)

los materiales necesarios para esta clase están descritos en la página principal de este repositorio.

se recomienda adquirir el kit de MCI electronics, disponible en [Mercado Libre](#) a 13.990 CLP.

coordinar con profesor si quieren comprar en grupo.

computadores y microcontroladores (45 min)

definición de computador:

- máquina digital electrónica programable con lógica y aritmética
- está compuesto de muchos elementos reemplazables: CPU, GPU, memorias, etc
- características: + velocidad, + tareas, + costo, + complejidad

definición de microcontrolador:

- máquina digital electrónica programable con lógica y aritmética
- computador pequeño que cabe en un único chip
- el chip incluye CPU, memoria, timer, puertos seriales, input/output
- características: - velocidad, - tareas, - costo, - complejidad

actividad: hacer una lista sobre tareas fáciles y difíciles para computadores y personas.

más info sobre personas pioneras en computación:

- [Alan Turing](#)
- [Ada Lovelace](#)
- [Programmed inequality](#), libro de [Mar Hicks](#)

programar semáforo en Processing (45 min)

usando la aplicación Processing 4.x, programamos un semáforo de 3 luces, usando variables para definir colores, duraciones de luces y posiciones.

la solución propuesta está en la carpeta asociada a esta clase.

actividad: cuántos números enteros de 8 bits caben en nuestro computador?

<https://www.wolframalpha.com/input?i=2+GB+%2F+8+bit>

<https://github.com/montoyamoraga/infinite-folders>

próxima clase (5 min)

- electricidad y magnetismo
- componentes eléctricos

- programar semáforo en Arduino
- construir circuito para semáforo