

Carrera: Licenciatura en Sistemas**Asignatura:** Introdutoria a Licenciatura en SistemasGuía de ALGORITMOS n° 2

Sintaxis pseudocódigo. Autómatas, comandos. Variables: definición y tipos. Estructuras de Control. Propositiones. Bloques. Prueba de escritorio.

Teoría de números: Números pares, primos, coprimos. Máximo Común Divisor. Algoritmo de Euclides. Mínimo Común Múltiplo.

Herramientas TIC

Diseño en bloques. <https://developers.google.com/blockly/>

<https://blockly-demo.appspot.com/static/demos/code/index.html?lang=es>

Pseudocódigo: ver archivo Introdutoria Sistemas - (Sintaxis pseudocodigo).pdf

Autómatas, comandos.

1. Diseñar un algoritmo para hacer una llamada por móvil, considerar que puede ser realizada por el prestador del servicio de telefonía móvil, whatsapp o skype
2. Diseñar un algoritmo para ver una película de un repositorio de contenido digital.

Variables: definición y tipos.

1. Diseñar un algoritmo que utilice dos variables una para nombre y otra para apellido, en una tercera variable concatenar nombre y apellido e imprimirla.
2. Diseñar un algoritmo que utilice dos variables una **a** y **b** calcule el resultado en la variable **r** luego imprima con este formato ej: 2+3=5
3. Crear una variable lógica *p* inicializada en true, crear una variable *q* inicializada en la negación de *p*, imprimir *p* y *q*.
4. Escribir el algoritmos que dadas las variables lógicas *p*, *q*, calcular *r* si es el resultado (*p* AND *q*). Imprimir *p*, *q* y *r*. Crear todos los escenarios posibles para los valores que pueden tomar *p* y *q*.
5. Escribir el algoritmos que dadas las variables lógicas *p*, *q*, calcular *r* si es el resultado (*p* OR *q*). Imprimir *p*, *q* y *r*. Crear todos los escenarios posibles para los valores que pueden tomar *p* y *q*.

6. Generar el algoritmo y los escenarios de prueba para demostrar:

$$NOT(p \text{ AND } q) \Leftrightarrow (NOT p) \text{ OR } (NOT q)$$

7. Generar el algoritmo y los escenarios de prueba para demostrar:

$$NOT(p \text{ OR } q) \Leftrightarrow (NOT p) \text{ AND } (NOT q)$$

8. Escribir el algoritmo para que dado un número determine si es par o impar. Realizar la prueba de escritorio para los escenarios posibles.
9. Escribir el algoritmo para que dado un número *a*, determine si es múltiplo de otro número *b*. Realizar la prueba de escritorio para los escenarios posibles.
10. Escribir el algoritmo para que dado un año determinar si es bisiesto.

Serán bisiestos los años divisibles por 4, exceptuando los que son divisibles por 100 y **no** divisibles por 400. Ejemplos: son bisiestos 1996, 2004, 2000, 1600; No son bisiestos 1700, 1800, 1900, 2100

11. Dados *a* y *b* escribir el algoritmo que imprima *a* y *b*, teniendo en cuenta previamente que si *a* es mayor que *b* intercambie el contenido de las variables.
12. Dados *a*, *b* y *c* escribir el algoritmo que imprima *a*, *b* y *c*, teniendo en cuenta previamente ordenar en forma ascendente el contenido de las variables.
13. Escribir el algoritmo para emitir un pre-ticket de compra de un artículo con su precio y la cantidad de unidades. Si la cantidad de artículos supera las 4 unidades se aplicará el 5% de descuento; por ejemplo: Si compra 6 paquetes de harina \$ 25.- el sub-total es \$150.- el Descuento (5%) es \$7,50 y el total pre-ticket: \$142,50.-
14. Escribir el algoritmo para emitir un pre-ticket de compra de un artículo con su precio y la cantidad de unidades. Calcular el descuento (3X2) si cada tres unidades de compra tendrá un descuento del importe de una unidad; por ejemplo: Si compra 7 latas de tomates a \$ 30.- el sub-total es \$210.- el Descuento (3x2) es \$60 y el total pre-ticket: \$150.-
15. Escribir el algoritmo para simular lanzar una vez un dado.
16. Escribir el algoritmo para simular una vez una moneda.