



UNIVERSIDAD
DE LA FRONTERA

Programación de Computadores

Guía de Ejercicios Scratch

Septiembre 2016



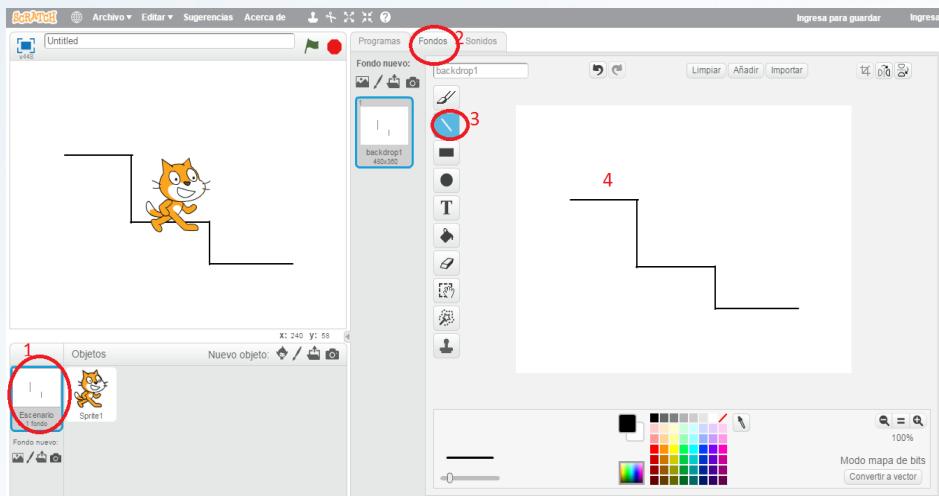
Departamento de
Computación e Informática

Ejercicios Resueltos

Ejercicio 1

Cree un circuito en donde un objeto o personaje al pasar muestre un mensaje y no se pueda devolver.

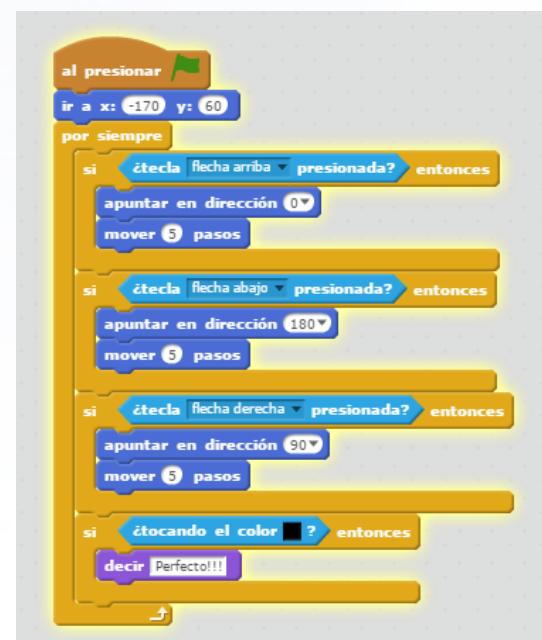
Primero debes crear el escenario con el circuito. Esto se hace en “Escenarios”. Luego pinta líneas o algo de un color.



Luego puedes cambiar el tamaño al personaje u objeto para que tenga más dificultad.

¡Ahora debes crear el programa!

Debes crear el evento para que empiece el programa y luego empezar las condiciones para que el objeto se mueva y comprobar si toca o no el circuito.



🖱 Ejercicio 2

A partir del ejercicio anterior, haga que muestre un mensaje si no está en el circuito.

Para esto solo debes cambiar la condición de si está tocando el circuito a un Si sino.



🖱 Ejercicio 3

Ahora haz que si deja de tocar el circuito pierda y termine el programa o muestre un mensaje.

Hay varias formas de hacer esto, una forma es editando el sí sino de la condición “tocando el color”.

Las otras formas las tendrás que hacer un poco más adelante. ¡No olvides este programa!



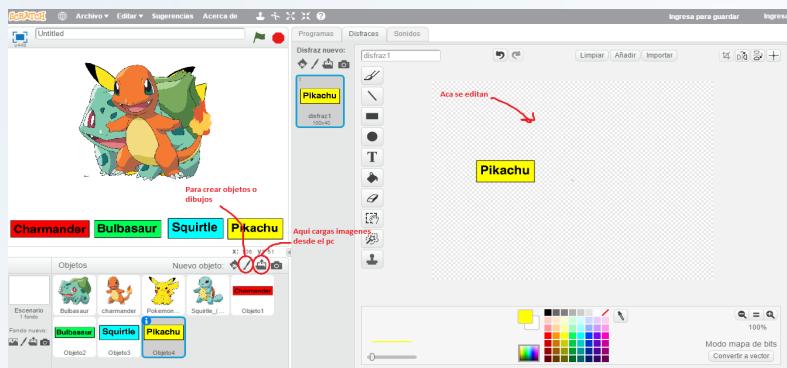
🖱 Ejercicio 4

Ahora sabiendo ciclos has alguna otra forma para detener el programa al salir del circuito.

🔗 Ejercicio 5

Crea 4 botones que al pasar el cursor muestren una imagen que le corresponda.

Primero busca imágenes o usa unas de la biblioteca de Scratch y añádelas al proyecto. Puedes crear las imágenes y los botones en el editor de Scratch



Luego, a cada “objeto” o imagen la puedes programar. En este caso lo primero es tener las imágenes de cada uno ocultas hasta que se pase por un botón, luego si se pasa por otro botón que este último se oculte y se muestre el otro y así...

Para hacer esto, debes tener en consideración la posición X e Y del cursor, para eso pasa el cursor por los límites del botón y en la esquina te muestra las coordenadas x,y.

Así quedarían todos, solo con diferencia en cada coordenada.



Son 2 condiciones, ya que no vale solamente que este en la posición correcta en X, sino que también debe estar en la posición Y correcta. Si se cumplen las 2 entonces mostrar la imagen.

El bloque “por siempre” sirve para evaluar en todo momento una vez ejecutado el programa o hacer algo en todo momento, si necesitamos saber algo en todo momento debe hacer uso del bloque “por siempre”. Ejemplos: saber si presionan un botón, si toco algo etc...

⚡ Ejercicio 6

A base del ejercicio anterior, ahora cree un programa en el cual al mostrar una imagen compruebe si esta correcta con el botón. (Ejemplo, Imagen: manzana, le corresponde el botón manzana.)

Consejos: Para esto puedes usar el mismo proyecto anterior, pero con un azar como este en cada imagen.

En donde el “1” cambia en cada imagen (Para que al ejecutar salga una distinta o al azar)... ¡Queda de tarea!

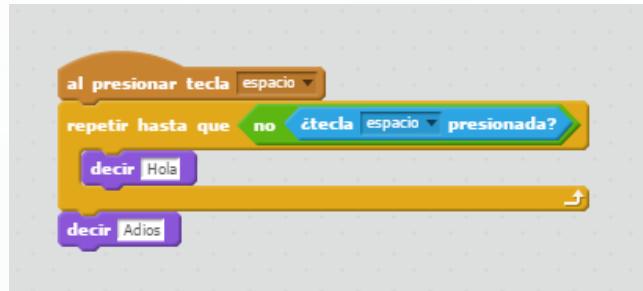


⚡ Ejercicio 7

Haga que el personaje u objeto muestre un mensaje al tener un botón presionado, en caso contrario otro mensaje.

Para hacerlo ahora usamos ciclos, en este caso “Repetir hasta que”

Entonces al presionar la tecla “espacio” se ejecutara el programa y el ciclo REPETIR A N VECES lo que este dentro HASTA QUE algo se cumpla en este caso QUE SE DEJE DE PRESIONAR LA TECLA.

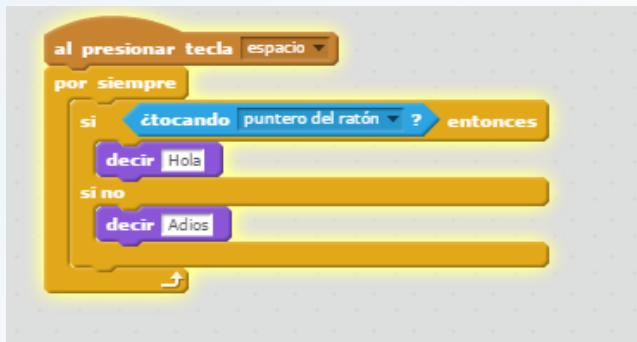


⚡ Ejercicio 8

Ahora si el cursor del mouse esta sobre el objeto que diga un mensaje.

En este caso usar un ciclo por siempre es lo mejor, así en todo momento este ciclo se está volviendo a ejecutar y preguntar si el cursor esta sobre él.

Luego con una condición dentro del ciclo puedes mostrar el mensaje.



⚡ Ejercicio 9

Mientras mantengas presionada una tecla que el objeto vaya sumando de 1 en 1, desde 0 hasta que lo dejes de presionar.

Para esto debes crear una variable.

Prueba sin usar el bloque “esperar 0.5 segundos”...

Esto pasa porque el computador lo procesa muy rápido, si no le dices que espere lo hará lo más rápido posible.

Luego se cambia el “número” por 1... esto le suma 1 cada vez que se repita el ciclo.



Ejercicios Propuestos

🖱 Ejercicio 10

Para demostrar sus habilidades gráficas utilizando Scratch, un individuo quiso hacer una animación (como el gatito) que bailara. Este puede bailar tres tipos de música: Pop, Street dance, y electrónica. El algoritmo debe entregar la opción de elegir el tipo de música y, en base a la elección debe existir una secuencia de pasos o movimientos que debe realizar el objeto.

🖱 Ejercicio 11

Un vendedor en un mall gana aproximadamente \$2.000 pesos por hora de trabajo. Si trabaja un periodo mayor a 40 horas por semana, las horas extras se pagan a \$3.000.

Si las ventas que realiza superan los \$300.000 a la semana, se le bonifica con un 5% del monto vendido, pero si sus ventas superan los \$500.000 a la semana, se le bonifica con un 10% del monto vendido.

Cree un programa en scratch que permita saber cuanto gana de forma semanal y mensual.

🖱 Ejercicio 12

Un día Alejandro estaba aburrido y no sabía cómo distraerse, así que se propuso ir desde el departamento de Ciencias jurídicas y empresariales hasta el departamento de Ingeniería de Computación e Informática, estaba buscando una mesa para sentarse cómodamente y prender su computador para jugar StarCraft. Desde el edificio en el que estaba podía pasar al edificio RA, R1, D, E, DIS. Al pasar por cada uno de éstos edificios se propuso lanzar una moneda, si salía cara iba a buscar una mesa en ése edificio, si salía cruz seguía caminando.

Se necesita que se programe su trayecto y decisión en Scratch.

🖱 Ejercicio 13

Se necesita un algoritmo en el cual se debe ingresar un número y permita encontrar los divisores de éste.

💡 Ejercicio 14

Ingresar dos números, identificar si son primos o no, debe mostrar en pantalla todos los números pares que estén entre esos dos números y realizar la secuencia de Fibonacci con el de mayor valor.

💡 Ejercicio 15

Se necesita un programa en Scratch en el cual se ingresa 3 notas de un ramo para un curso de n alumnos (definidos por el usuario), se debe sacar el promedio y contabilizar a los alumnos que han aprobado, estén en situación de dar examen y a los que han reprobado.

* Las notas se pueden generar de manera aleatoria, y deben ser de 1 a 7.

💡 Ejercicio 16

Para calcular la N-sima suma de una serie, es necesario identificar el comportamiento de esta. Calcule la suma de los primeros 100 y los primeros 200 términos de la serie:

$$1 + \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots$$

💡 Ejercicio 17

En una clase de calculo solicitan graficar una serie de funciones cuadráticas de la forma $ax^2 + bx + c = 0$, para visualizar su comportamiento. Cree un programa en Scratch que permita graficar una ecuación cuadrática ingresando los coeficientes (a,b,c).

✍ Ejercicio 18

Cree un reloj “análogo” con que muestre horas, minutos y segundos. (utilice la imagen como referencia)



✍ Ejercicio 19

Cree un programa que muestre los números primos entre 200 y 500

✍ Ejercicio 20

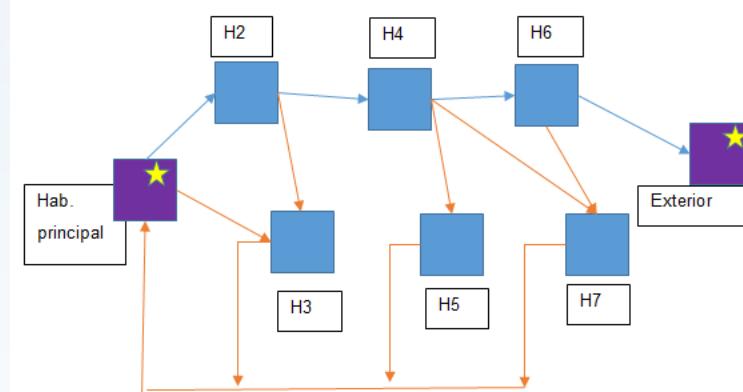
Hacer una calculadora que pueda realizar las operaciones básicas como sumar, restar multiplicar y dividir y que, además, dado dos puntos en el plano pueda sacar la pendiente de la recta que los une, y que tenga también la funcionalidad de, dados dos puntos, conocer la distancia entre ellos. El programa puede realizar estas operaciones n veces, o hasta que el usuario se canse.

💡 Ejercicio 21

Crear un juego basado en la siguiente descripción:

"Por curioso te has metido en un laberinto adentro de una casa embrujada, cada habitación de la casa posee al menos dos puertas, sólo hay una ruta para salir al exterior y si te equivocas de puerta vuelves al principio. No puedes regresar a la habitación anterior"

El juego acaba cuando se logra escapar.* No es necesario preguntar nada en las habitaciones 3,5 y 7 ya que automáticamente se debe devolver a la habitación principal.



💡 Ejercicio 22

Utilizando el juego anterior, ahora se debe complejizar un poco más, agregue dos habitaciones por las que se pueda llegar desde la habitación principal al exterior, pero en cada una de las habitaciones puede aparecer un fantasma con una probabilidad de 50%, se recomienda utilizar números aleatorios entre 0 y 1, si aparece un fantasma el juego termina.

💡 Ejercicio 23

Crear un juego basado en la siguiente descripción:

"Es el momento en que los zombies inundan el mundo, tú eres un científico y explorador experto, quien ha encontrado un compuesto que podría eliminar a los monstruos y devolver por fin la paz a las personas, tu misión es transportarte desde tu refugio base al laboratorio dónde se crea la cura, que se encuentra tres ciudades más al norte, pero, para tu mala suerte te has quedado sin municiones en las armas que llevabas".

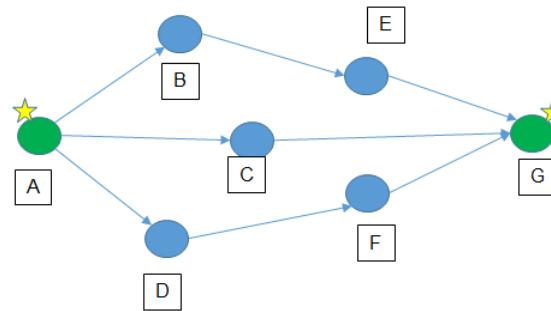
Para crear el juego se debe basar en el mapa de la figura.

Se debe viajar por una de las rutas delimitadas por los nodos en la dirección de las flechas, por ejemplo, A-B-E-G.

La forma de llegar seguro al destino es no encontrarse con ningún zombie.

Se debe preguntar al jugador cual camino desea tomar. Los zombies aparecen de forma aleatoria según lo siguiente:

1. En la ciudad C, la cantidad de zombies puede ser un número aleatorio entre 0 y 3 (0= no hay zombies, 1= hay 1, 2= hay 2 y 3 = hay 3).
2. En el resto de ciudades la cantidad de zombies es un número aleatorio entre 0 y 1 (0 = no hay zombies, 1 = hay por lo menos un zombie)
3. Las ciudades A y G están libres, son zonas seguras.
4. Gana el juego la persona que no se haya encontrado con ningún zombie en su camino, si encuentra al menos uno, se queda sin cerebro.



🖱 Ejercicio 24

El hijo del dueño de una pizzería estaba tan fascinado con los conocimientos adquiridos en el ramo de Programación de Computadores, que decidió crear un prototipo en la plataforma Scratch para registrar y contabilizar el valor de cada una de las pizzas que se pedían. La masa básica con queso y salsa de tomates cuesta \$3.200 en tamaño familiar y \$2.400 en tamaño mediano.

La pizza puede ser vegetariana o no, la vegetariana puede llevar champiñones, que aumentan el valor \$1.200, o carne de soya (\$1.000). Mientras que la normal, puede llevar Jamón (\$1.000), salame (\$1.000) o una mezcla de ambos (\$1.500).

Es posible también agregarle una mezcla de verduras y vegetales, hay tres opciones:

1. Tomate, choclo, pimiento verde (\$1.500)
2. Cebollas fritas, pimiento rojo, tomate, choclo (\$2.500)
3. Palmito, pimiento verde (1.000)

Se puede agregar aceitunas por \$300, y a las pizzas que no son vegetarianas se les puede agregar champiñones por \$500, pero es una porción menor.

Al final el programa debe entregar el resumen de las características de la pizza y el valor.

🖱 Ejercicio 25

Un pastelero italiano conoce una gran cantidad de recetas de distintos pasteles, decidió pedirle a un estudiante que le diseñara un algoritmo para automatizar la petición de requerimientos de los clientes.

Se pueden hacer pasteles con biscocho de vainilla, chocolate o red velvet, cualquiera cuesta \$3.000 (biscocho de 15 personas) y \$5.000 (bizcocho de 25 personas), y es posible rellenar con dulce de leche, frutas, queso, de pueden elegir dos rellenos distintos, cada uno tiene un valor de \$1.500, y si lleva distintos, cada uno cuesta \$1.000 y, para cubrir se puede con crema chantilly (\$2.000) o fondant (\$3.000).

Se puede decorar con chocolate (\$1.000), frutas o alguna salsa (\$800) y, además se podría hacer una dedicatoria que sería gratis.

Todo lo anterior también puede estar en versión para diabéticos, pero con esta versión no se puede cubrir con fondant.

Se debe entregar el resumen de las preferencias del pastel al cliente y el valor de éste.

🖱 Ejercicio 26

Una tienda de colchones vende distintos tipos, posee una línea normal para su uso en domicilios y una línea de hoteles cuyos colchones tienen un diseño de costura superior distinto y están fabricados con un resorte especial. Ambas líneas se fabrican en medidas de una plaza y dos plazas. Se necesita hacer un sistema de venta de colchones.

Los valores de venta son:

1. Colchón normal 1 plaza \$94.990.
2. Colchón normal 2 plazas \$154.990.
3. Colchón 1 plaza línea hotel \$129.990
4. Colchón 2 plazas línea hotel \$209.990

Sólo los clientes particulares pueden pagar con una tarjeta de un retail asociado, CMD, obteniendo con ello un 8% de descuento en la compra y acumulando 1 punto por cada \$100 pesos de compra.

Si es un hotel, obtiene descuento dependiendo de la cantidad de dinero invertido, si compra más de \$200.000, obtiene un 10% de descuento, si compra entre \$250.001 y \$300.000 tiene un 15% de descuento y si la compra es mayor a \$300.000 el descuento es de 20%

Al finalizar la transacción el programa debe mostrar el valor total de la compra, en caso de que se haya hecho descuento, el total de éste y si se pagó con tarjeta, los puntos acumulados. Además, debe preguntar si se desea realizar otra transacción.

🖱 Ejercicio 27

Suponga un tablero de 15 casillas, por las cuales se mueve una ficha de acuerdo al número que se obtiene al lanzar un dado. El movimiento está regido por las siguientes reglas:

- 1.- La ficha avanzará la cantidad de casillas indicadas por el dado, si el número obtenido es Par.
- 2.- La ficha retrocederá la cantidad de casillas indicadas por el dado, si el número obtenido es Impar.

El juego terminará cuando la ficha recorra completamente el tablero. Además, el algoritmo deberá indicar la cantidad de jugadas realizadas.

🔗 Ejercicio 28

Se está desarrollando una carrera en la que participan 5 competidores, a cada uno de ellos se le debe ingresar el nombre, deben correr 3 competencias, 100 metros planos, doscientos y trescientos. Para calcular lo anterior, a cada uno de los concursantes se les debe generar un número aleatorio que correspondería al tiempo en el que cada uno logró llegar a la meta, gana cada competencia el que llega primero y deben ser mostrados los 3 primeros lugares.

- Nadie puede llegar a menos de 9 segundos en ninguno de los eventos, debe ser un registro física y biológicamente correcto, como también, nadie debería llegar horas más tarde a la meta.

** No es necesario desarrollar gráficamente la carrera.

🔗 Ejercicio 29

Un día Ash salió a buscar pokemones, para ganar por fin su próximo torneo decidió encontrar sólo a algunos específicos que le marcarían una ventaja el día que los necesitara. Para ello organizaría su recorrido según un algoritmo en Scratch:

1. Buscaría principalmente a seres de nivel 25 o superior.
2. Si era un Magikarp no lo tomaba. A menos de que pasara el nivel 19 (ya que al nivel 20 evoluciona y es más fácil de atrapar que Gyarados).
3. Cuando le queden sólo 2 pokebolas vacías, los pokemon tipo planta tiene prioridad ya que su próximo rival tiene tipo roca.

Se debe suponer que se tienen solo 6 pokebolas y no se puede adquirir otra hasta volver a la casa. (*Basado en juegos de consola).

Realice el juego con al menos 3 pokemones distintos, no es necesario la recreación grafica.

🔗 Ejercicio 30

Suponga que se le pide construir un programa para un matadero, el cual debe ser capaz de ingresar, para una cantidad X de animales vendidos, el peso de cada uno y sobre esto, mostrar en pantalla los siguientes informes:

Cantidad de Animales vendidos

1. Valor total transado
2. Cantidad de animales por categoría (según tabla anexa)
3. Valor transado por categoría (según tabla anexa)
4. La comisión si por cada venta recibe el 10%
5. Ganancia del matadero.

Considere que:

Categoría	Condición	Precio de Venta (\$/Kg.)	Costo por Animal
A	Peso \geq 500 Kg.	\$ 250 por Kg.	\$ 50 por Kg.
B	250 Kg. \leq Peso < 500 Kg.	\$ 200 por Kg.	\$ 70 por Kg.
C	Peso < 250 Kg.	\$ 150 por Kg.	\$ 30 por Kg.

🔗 Ejercicio 31

Considere a una persona que va a un casino con 20 fichas para apostar a los dados. Este jugador tiene por cábala apostar siempre a los números pares. Las reglas del juego son las siguientes:

- 1.- En cada jugada se apuesta como máximo una ficha
- 2.- Si sale un número par el jugador gana una nueva ficha
- 3.- Si sale número impar el jugador pierde su ficha.

El juego terminará cuando el jugador pierda todas sus fichas. Además, deberá indicar en cuantas jugadas perderá todo el dinero.

Ejercicio 32

Considere una empresa que se dedica a administrar las inversiones de sus clientes en la bolsa de Santiago. Esta empresa ofrece a sus clientes 3 opciones de inversión a dos niveles de riesgo (alto y bajo). Cada cliente de la empresa debe indicar el tipo y la cantidad de acciones a comprar. Su algoritmo debe ser capaz de administrar las inversiones de X clientes. Además, el algoritmo debe mostrar en pantalla los siguientes informes al finalizar el periodo:

Cantidad Total de Acciones Vendidas

Valor Total de Venta

Detalle por cada tipo de Acción:

Cantidad Total de acciones vendidas

Valor total de venta

Detalle por cada tipo de Riesgo:

Cantidad Total de acciones vendidas

Valor total de venta

Utilidad total de los clientes de la empresa

Utilidad total de la empresa

Considere que:

Tipo Acción	Valor Acción	Rentabilidad (% Inversión)	Comisión (% Utilidad)
A			
Riesgo alto	\$ 250 por Acción.	30%	5%
Riesgo bajo	\$ 200 por Acción.	15%	2%
B			
Riesgo alto	\$ 200 por Acción.	25%	3%
Riesgo bajo	\$ 150 por Acción.	12%	1%
C			
Riesgo alto	\$ 150 por Acción.	20%	1%
Riesgo bajo	\$ 100 por Acción.	10%	0.5%

🖱 Ejercicio 33

Construya un programa que le permita simular el juego de azar «Loto». Para este juego tenga en cuenta las siguientes consideraciones.

1. Se sortean seis números, entre 1 y 36 no repetidos.
2. El usuario compra un cartón con seis números, entre 1 y 36. No puede repetirse ningún número en un cartón.
3. Para ganar el Loto, el jugador necesita tener las seis coincidencias con los números sorteados.
4. Los premios a repartir son: 6 coincidencias \$ 100.000.000; cinco coincidencias \$ 50.000.000; cuatro coincidencias \$ 5.000.000; menos de cuatro coincidencia no obtiene premio

Su programa deberá indicar por cada juego la cantidad de coincidencias y el premio ganado. Se recomienda el uso de listas.

🖱 Ejercicio 34

Construya un algoritmo que represente un juego de mesa para dos jugadores. El juego consiste en avanzar sobre un tablero de 30 casillas la cantidad de espacios que indica el lanzamiento de dos dados. Ganará el jugador que llegue primero a la última casilla. Tenga en cuenta las siguientes consideraciones:

1. Los lanzamientos de cada jugador son por turnos intercalados, es decir, primero el jugador 1, luego el jugador 2, luego nuevamente el jugador 1 y así sucesivamente.
2. Cada vez que en un lanzamiento salga el mismo número en ambos dados, el jugador tiene derecho a una nueva jugada inmediatamente.
3. Cada vez que el jugador caiga en una casilla múltiplo de 10 puede avanzar automáticamente 3 espacios.
4. Cada vez que el jugador caiga en una casilla múltiplo de 7, debe retroceder 2 espacios.
5. Cada vez que un jugador caiga en una casilla ocupada por el otro jugador debe retroceder 1 espacio.
6. La llegada debe ser con los números exactos, si se pasa deberá regresar los espacios que sobren.

Por cada jugada debe indicar el resultado del lanzamiento de los dados y la casilla a la que avanza. Al finalizar, debe entregar que jugador ganó y en cuantas jugadas lo hizo.

✍ Ejercicio 35

Suponga que el ministerio de salud está realizando un catastro acerca de la nutrición de los estudiantes básicos de los colegios de la ciudad de Temuco. Suponga que Temuco, posee X colegios. Por cada estudiante evalúa el peso y la estatura. Construya un algoritmo que entregue el promedio del peso, la estatura y el Índice de masa corporal para cada colegio. Si se considera la tabla adjunta, indique por cada colegio cuantos estudiantes tienen en cada categoría.

Al finalizar, como resumen general, muestre la cantidad y porcentaje de estudiantes por cada categoría del IMC.

Nota:

Considere que el IMC se calcula de la siguiente manera: $IMC = \frac{Kg}{mts^2}$

Índice de Masa Corporal (IMC – Kg/m ²)	Categoría
$IMC < 20$	Desnutrición
$20 \leq IMC < 25$	Normal
$25 \leq IMC < 30$	Sobre-Peso
$30 \leq IMC < 35$	Obesidad Leve
$35 \leq IMC < 40$	Obesidad Moderada
$IMC \geq 40$	Obesidad Mórbida

🖱 Ejercicio 36

Construya un programa que genere una lista a partir de otras dos listas. Esta tercera lista debe contener los datos de las dos primeras de manera intercalada, es decir, primero el primer dato de la primera lista, luego el primer dato de la segunda lista, luego el segundo dato de la primera lista y así sucesivamente (Imagine que baraja dos mazos de naipes). Las dos primeras listas deben ser creadas con una longitud indicada por el usuario (no necesariamente iguales) y llenadas con números enteros aleatorios entre 1-15.

Debe mostrar el contenido de las tres listas.

🖱 Ejercicio 37

Considere el caso inverso al ejercicio anterior, es decir, en esta ocasión debe construir dos listas a partir de una lista inicial. Esta primera lista, contendrá N datos, generados aleatoriamente (entre 1 y 15). En una lista se guardarán todos los datos pares que contenga la primera lista, en la otra, todos los datos impares.



UNIVERSIDAD DE LA FRONTERA

Guía de Ejercicios de Scratch

UNIVERSIDAD DE LA FRONTERA

FACULTAD DE INGENIERÍA Y CIENCIAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

Avda Francisco Salazar 01145
Temuco – Chile / casilla 54-D
Fono (56) 45 2325000 /2744219

dci.ufro.cl



Departamento de
Computación e Informática