

NOMBRE DE LA CLASE:

Condicionales (Programando con tarjetas)

Duración: 45 - 60 minutos : Preparación: 2 minutos

Meta: Esta clase introduce los condicionales, especialmente bucles y sentencias “if”.

RESUMEN:

La clase trata de un juego. Después de introducir la idea de condicionales, sentencias “si”, y bucles, la clase puede practicar en conjunto el uso de un programa ya escrito, con un mazo de tarjetas y suerte.

OBJETIVO:

Los estudiantes —

- Ganarán experiencia en la determinación de resultados (verdaderos o falsos) de los condicionales
- Evaluarán expresiones lógicas para determinar qué rama del programa seguir

MATERIALES:

- Un conjunto de páginas de programación por clase (Esto tiene como al menos un ejemplo de programa incluido)
- Una baraja francesa

PREPARACIÓN:

Separar del mazo las cartas con figuras del resto. Los comodines pueden permanecer

entre las cartas normales para usarse en eventos especiales, como borrar todos los puntos, o dejar que el equipo gane instantáneamente.

Mostrar una de las páginas de programación a la clase.

VOCABULARIO:

Condición—Una sentencia que es verdadera o falsa dependiendo de la situación

Sentencia “sj”—Una línea que determina si se va a ejecutar o no una determinada porción de código

Sentencia “sino”—Otra forma de decir “en caso contrario”

Funciones—Una porción de código que puede ser llamada una y otra vez

Incremento—Añadir cierta cantidad (generalmente 1), una o muchas veces

Decremento—Restar cierta cantidad (generalmente 1), una o muchas veces

Sentencias anidadas—Una sentencia que se encuentra dentro de otra sentencia

REPASO:

La intención de este repaso es recordar lo visto en la clase anterior. Si cubres las actividades en distinto orden, por favor realiza tu propio repaso aquí.

Preguntas para la participación en clase:

- ¿Qué hicimos en la clase anterior?
- ¿En qué difiere el bloque “mientras” del bloque “si”?

Debate:

- ¿Se imaginan una forma de representar cualquier situación de la vida real con un bucle “mientras”?



Los condicionales son la forma que tienen las computadoras de tomar decisiones.

DESARROLLO:

Este juego tiene el potencial de ser algo que puedas repetir una y otra vez. Es infinitamente personalizable, ajustable a cada edad, experiencia previa y cantidad de estudiantes. Esta clase cubrirá el juego para chicos de 7 a 9 años, pero siéntete libre de adaptarlo como te guste mientras mantengas el espíritu de la actividad.

Para empezar, los estudiantes necesitan refrescar sus conocimientos sobre condicionales, incluso cuando hayan visto las actividades en línea con Blockly. Puede ayudar empezar con ejemplos de la vida real:

**“Si levantas tu mano, voy a aplaudir.
Sino, voy a tocar mi nariz.”**

A continuación, toca tu nariz hasta que la clase levante su mano, y luego aplaude para dar a entender que están cumpliendo con la condición.

“¿Qué ha pasado aquí?”

Los estudiantes pueden intentar comentar los eventos que se les ocurra, para que puedas guiarlos hacia el camino que pretendes.

“¿Te di una “condición”, cierto? Como dije, ante una condición deberíamos hacer una cosa, si esa condición no se dió, haría otra cosa. Los condicionales son la forma que tienen las computadoras de tomar decisiones.”

Luego, pueden intentar otro ejemplo. Esta vez, introduce la parte del “sino”.

**“Si todos ponen sus manos sobre el escritorio, entonces voy a rascarme la cabeza.
Sino, voy a poner una mano sobre mi cadera.”**

Esto suena bastante similar al primer caso por lo que deberían poder inferir lo que la parte del “sino” está diciendo, pero si tienen problemas, explicar que ese “sino” es similar a decir “en caso contrario”.

“Buen trabajo. Ahora hagamos algo interesante. Voy a dejar esta pila de cartas aquí. Y vamos a jugar a un juego.

**Si saco una carta roja, obtengo un punto.
Sino, ustedes tienen un punto.”**

Escribir esta oración en el pizarrón, para que sea posible apuntar a cada caso cuando suceda. Saca algunas cartas de la pila, para añadir puntos a quien corresponda. Cuando la carta sea roja muestra que cumple con la condición, y cuando sea negra, muestra que coincide con el caso del “sino”. Después de algunas movidas (o cuando los estudiantes ya se sientan cómodos) decir que el juego terminó, y prepararse para avanzar de nivel.

“Bueno, vamos a usar esta idea de las condiciones, para hacer un juego de “condicionales”. Llamamos a esto condicionales de la sentencia “si”, porque hay una condición

puesta en esas sentencias. Algo es verdadero o no lo es. Si es verdadero, entonces ejecutamos la instrucción dentro de la sentencia “si”. En caso contrario, ejecutamos la instrucción dentro del “sino”.

Dile a la clase sepa que va a jugar un juego. El juego dura 4 turnos de cada lado. La clase debe dividirse en “lado izquierdo” y “lado derecho” (si consideras que esto es un problema, pueden dividirse entre los estudiantes vs el docente, y reemplazar “equipo.puntos += 1” por “docente.puntos += 1”). Van a competir por conseguir la mayor cantidad de puntos.

Para que las reglas queden en claro, escríbelas en el pizarrón. El código representado a continuación es la forma en que los condicionales son escritos en Javascript, el lenguaje de programación más usado en el mundo. En las actividades en línea de este curso, hemos escrito código usando Blockly, que usa bloques visuales para representar código. El código que hemos escrito en esas actividades puede ser también representado en otros lenguajes de programación. Por ejemplo, en nuestras actividades en línea, si cliqueas en la parte de “Mostrar código”, en la pestaña púrpura, pueden los bloques del editor representados en Javascript. El siguiente código es para la actividad que haremos el día de hoy:

```
1) if (carta.color == negro) {  
2) equipo.puntos += 1;  
3) }  
4) else {  
5) equipo.puntos -= 1;  
6) }
```

Querrás repasar este código línea por línea, ya que la mayoría de los estudiantes probablemente lo vean como algo nuevo.

1) Esta línea es tu sentencia “si” (“if”, en inglés). Es la línea que nos permite saber que si el color de la carta es igual a negro, entonces haremos algo. ¿Notas que hay dos signos de igual? En muchos lenguajes de programación esta es la forma en que verificamos por igualdad, no asignando que algo es igual a otra cosa. Esta línea termina con una “llave que abre”. Esto nos permite saber que en la siguiente línea le pertenece a la sentencia “si”.

2) Esta línea dice literalmente “Puntos más igual uno”. Significa que tomas los puntos y les sumas un punto.

3) Esto es una “llave que cierra”. Dice que la sentencia “if” ha terminado.

4) Aquí empieza la sentencia “sino”. Al final de esta línea veremos otra llave que abre. Esto significa que este es el comando de lo que se ejecutará sabiendo que la condición de la sentencia “si” no es verdadera.

5) Esta línea es similar a la línea 2, pero con una diferencia. ¿Te das cuenta de cuál? ¿Qué se supone que hace? Significa que cualquiera sea la cantidad de puntos que tienes, le resta uno a eso.

6) Finalmente, en la línea 7, tenemos la última “llave que cierra”. Nos dice que la sentencia “sino” ha terminado.

Aquí el código en Javascript junto al mismo código representado en pseudocódigo, que es una representación informal de escritura de programas que es fácil de leer. Cada línea de pseudocódigo explica lo que hace el código Javascript, paso a paso.

Javascript	Pseudocódigo
if (tarjeta.color == negro) { equipo.puntos += 1; }	Si el color de la carta es negro, entonces los puntos del equipo se incrementan en 1
else { equipo.puntos -=1;; }	sino los puntos del equipo se decrementan en 1

Después de explicar esto, es bueno volver hacia tras y ver el panorama general. ¿Que significa? ¿Podemos expresarlo con palabras? **Esencialmente dice que si una persona toma una carta negra, entonces su equipo obtiene un punto, sino pierde un punto. ¿Deberíamos empezar?**

Elige ocho voluntarios, cuatro de cada equipo, para ser los “programadores”. (Asumiremos que el juego es jugado de 20 a 40 personas, pero si tienes menos estudiantes, chequea los ajustes al final de la planificación de la clase).

Haz que cada uno de los programadores saque una carta. Esto no es un requisito, pero sería divertido ver qué equipo va primero (usando las reglas del final de la planificación). Con uno de los estudiantes no programadores agarra una carta numerada y pásala boca a bajo a los estudiantes sentados. Esto permite que todos se sientan involucrados con el juego, incluso si sólo queda tiempo para pocas movidas.

Envía al primer programador a elegir a un estudiante con una carta numerada. El estudiante debe levantar su carta. **¿De qué color es? ¿Con qué línea coincide? ¿Cuáles son los puntos de tu equipo?**

Cambia a un programador del equipo contrario, y repite esto hasta que todos los programadores hayan terminado su turno. **¿Cuál es el puntaje final? ¿Cuál es el equipo ganador?**

Si el juego se termina rápido, elige otro programa y empieza de nuevo. Si fue “realmente” bien, el desafío de la clase puede ser ayudar a modificar el programa para proveer más opciones. Una vez que lo jugaron unas cuantas veces, intenta explicar otras herramientas como bucles “mientras” y “sentencias anidadas”.

AJUSTES:

Edad

De 5 a 6 años: Representar gráficamente el programa puede ayudar. Tal vez quieras evitar que los programas escritos se vean como en Javascript. Mantiene simples a los algoritmos y cambia cosas sólo cuando los estudiantes entienden realmente cómo funcionan las reglas. El foco a esta edad es enseñar a pensar de manera lógica más que a leer programas.

De 7 a 9 años: Este grupo debería funcionar de igual manera a lo descrito en esta planificación. Gradualmente incrementar la complejidad del programa.

De 10 a 12 años: Los estudiantes pueden trabajar en grupos pequeños, unos contra otros. Pueden levantar cartas de la pila, en lugar de llevar las cartas a los escritorios, e incrementar la dificultad de los programas. ¡Trata de incentivar a los estudiantes a crear sus propios programas!

Experiencia

Principiante: Trata de mantener un condicional a la vez. Puede ayudar ver a los programas como imágenes o diagramas en lugar de Javascript.

Intermedio: Introduce código escrito en Javascript para dar una idea de cómo se ven los lenguajes de programación basados en texto. Si te sientes cómodo con más conceptos de Javascript, considera incluir estructuras de objetos como `estudiante.carta` o `carta.color` o trabajar en sentencias anidadas, y posiblemente incluso funciones: `carta.valor.doble()`

Experto: Gasta menos tiempo trabajando como clase, y más tiempo trabajando en grupos. Permite que los estudiantes escriban el programa, y dales desafíos como los siguientes:

- Crear un programa donde el as negro duplica tu puntaje
- Hacer un programa en donde haya distinto resultado con pares e impares, así como negra y rojas.

Cantidad de jugadores

1: Esto puede ser visto como una situación solitaria. Los estudiantes toman una carta de la pila, y comparan las cartas como un programa solitario, o uno que crearon ellos mismos. Un ejemplo podría ser que el estudiante siga un programa que requiere obtener cierto puntaje dentro de una cantidad de movimientos para ganar. Otro ejemplo puede ser que el estudiante juegue contra la “máquina”, donde los puntos acumulados distinto para entidades y el ganador es aquel que tiene más puntos al final del juego.

2-6 (par): Es considerado un grupo pequeño. Los estudiantes se mueven en turnos rápidamente por lo que involucrar a los que están sentados es más una distracción que un beneficio. En grupos pequeños, arma dos equipos y deja una pila en el centro del aula. Deja que los estudiantes tomen turnos de la pila central. No hay necesidad de tomar programadores voluntarios, ya que en esta versión todos pueden tomar cartas.

3-7 (impar): Juega igual que en lo descrito anteriormente, pero con la salvedad de un “Programa contador”, que será la persona que recuerda los puntajes para tomar la decisión final sobre qué líneas del programa son relevantes.

8+: Esto funciona bien como una clase tradicional. Si los estudiantes son más avanzados, entonces deja la pila enfrente del aula, en lugar de elegir cartas del escritorio de alguno.

REGLAS:

Elige 8 voluntarios (4 de cada equipo) para ser los “programadores”

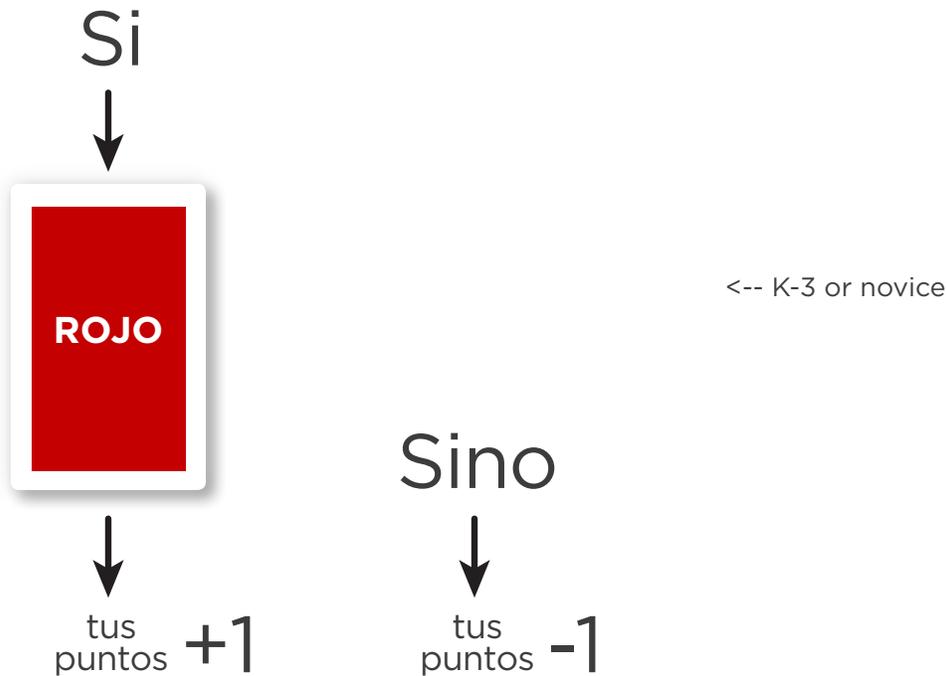
Ten voluntarios para elegir cartas para ver quién va primero.

Juega en un orden como el siguiente:

Reina, Rey, Jack - (Corazones, Diamantes, Tréboles, Picas) de cada una

Example:

- Reina de Corazones
- Reina de Diamantes
- Reina de Tréboles
- Reina de Picas
- Rey de Corazones
- Rey de Diamantes
- Rey de Tréboles
- Rey de Picas
- Jack de Corazones
- Jack de Diamantes
- Jack de Tréboles
- Jack de Picas



<p>Programa de ejemplo: De 8 años en adelante + Intermedio</p>	<pre>while (carta.valor < 5) { puntos ++; } // Este programa permite al jugar levantar cartas hasta // que la carta elegida sea mayor a 4 // Cada vez que el jugador levanta una carta, suma un // punto, independientemente del valor de la misma</pre>
---	---

<p>Ejemplo de programa: De 11 años en adelante + Más que intermedio</p>	<pre>while (carta.valor < 5) { equipo.puntos ++; if (carta.valor == 3) { equipo.puntos = equipo.puntos + 2; if(carta.palo == "treboles"){ otroEquipo.puntos += 2; } } } // Este programa permite al jugador elegir una carta // hasta que sea mayor a 4 /* Cada vez que el jugador levante, obtiene un punto, si la carta es un 3, entonces también obtiene otros 2 puntos, y si el palo era "treboles", le suma dos puntos al equipo contrario. */</pre>
---	---