



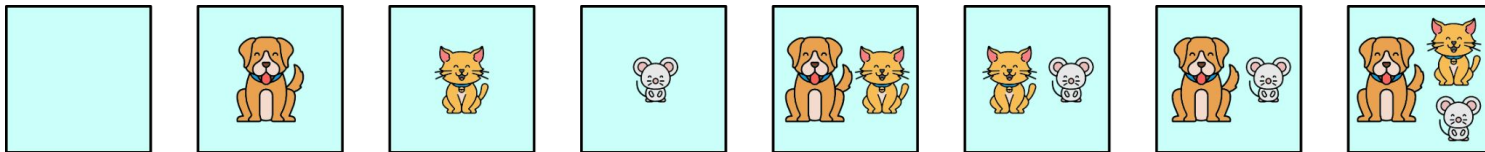
# Definición de expresiones

Teoría 3.B.

## Ejercicio simple

Se pide que identifique en qué escenarios la siguiente expresión es verdadera.

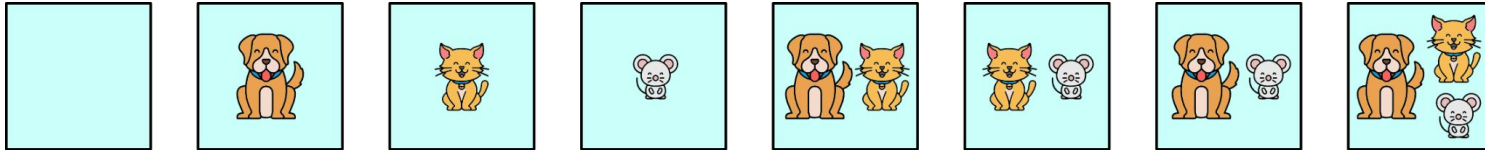
$$(\text{¿hay perro?} \wedge \text{¿hay gato?} \wedge \neg \text{¿hay ratón?}) \vee (\text{¿hay gato?} \wedge \text{¿hay ratón?} \wedge \neg \text{¿hay perro?}) \vee (\neg \text{¿hay gato?} \wedge \text{¿hay ratón?} \wedge \text{¿hay perro?})$$



## Ejercicio simple

Intentemos nuevamente, ahora con esta expresión.

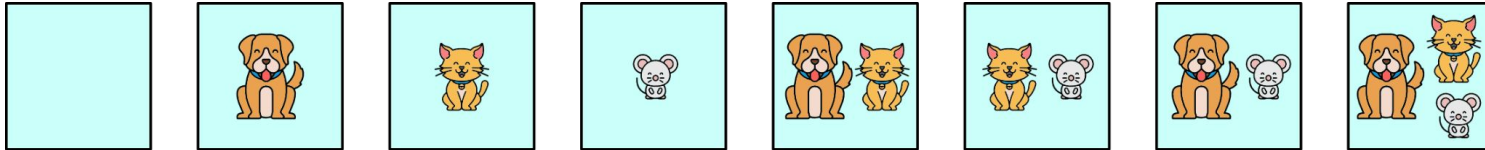
¿hay perro y gato solos?  $\vee$  ¿hay gato y ratón solos?  $\vee$  ¿hay perro y ratón solos?



## Ejercicio simple

Una más...

¿hay solo dos animales?





# ¿Comunicar a la máquina o a las personas?

Cuando exponemos todas las conectivas y los sensores básicos, ¿En quién pensamos? ¿Estamos pensando en seres humanos que van a leer eso? ¿O estamos pensando en satisfacer a la máquina?

Queremos que las expresiones comuniquen adecuadamente la idea que representan, igual que lo hacen los comandos.

En el mundo de los comandos, creábamos procedimientos y les dábamos nombres adecuados para que comuniquen bien las ideas.

¿Se puede hacer lo mismo con las expresiones? Spoiler, claro que se puede.

# Definiendo expresiones

---



## Definiendo expresiones

Hay que recordar que, a diferencia de los comandos que describen acciones, las **expresiones describen datos**.

La definición de una nueva expresión, va a estar dada en términos de equivalencia entre la nueva expresión que estamos definiendo, y expresiones de las que ya disponíamos. Ej.

**Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge$   $\neg$  ¿hay ratón?**

La gracia es poder dar mejores nombres a grupos de expresiones compuestas de varias partes y conectivas.



# Nombrando nuevas expresiones

Como vamos a definir únicamente expresiones de valores de verdad, la regla es simple. El **nombre** debe contener **signos de interrogación**, y corresponderse a una pregunta que se responda con **sí o no**.

No son verbos, ya que no hay acciones involucradas en una expresión. Es solo una pregunta, que contestamos en términos de otras preguntas.

Deben estar relacionados a lo que se quiere responder, y debe poder leerse claramente por una persona que no conoce el contexto del problema.






## Sintaxis usada

Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge \neg$  ¿hay ratón?

## Sintaxis usada

**Definir** ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge \neg$  ¿hay ratón?



Comenzamos por la palabra clave “Definir”, ya que al igual que con los procedimientos, estamos definiendo una cosa que antes no existía.


## Sintaxis usada

Luego el nombre, que como dijimos, es una nueva pregunta.

Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge \neg$  ¿hay ratón?

## Sintaxis usada


Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge$   $\neg$  ¿hay ratón?



El signo igual indica que la pregunta que definimos, es equivalente a realizar las otras preguntas de la derecha.

## Sintaxis usada

Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge \neg$  ¿hay ratón?



Finalmente, indicamos a qué es equivalente la pregunta que acabamos de definir. Deberían ser al menos dos partes unidas por conectivas.



## Sintaxis usada

Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge \neg$  ¿hay ratón?

Consiste en una definición de equivalencia, que se lee de izquierda a derecha.



## Sintaxis incorrecta

**$\text{¿hay perro?} \wedge \text{¿hay gato?} \wedge \neg \text{¿hay ratón?} = \text{Definir ¿hay perro y gato solos?}$**

La definición debe leerse de izquierda a derecha. Sí la palabra definir se pone del lado derecho de la igualdad, se pierde, y no se entiende la definición. Por más que haya un igual, no funciona como en matemática, donde se pueden cambiar los elementos de lado.

## Múltiples definiciones

Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge$   $\neg$  ¿hay ratón?

Definir ¿hay perro y ratón solos? = ¿hay perro?  $\wedge$   $\neg$  ¿hay gato?  $\wedge$  ¿hay ratón?

Definir ¿hay ratón y gato solos? =  $\neg$  ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge$  ¿hay ratón?

Podemos realizar varias definiciones, para distintas expresiones, según lo necesitemos.



## Múltiples definiciones

Definir ¿hay perro y gato solos? = ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge$   $\neg$  ¿hay ratón?

Definir ¿hay perro y ratón solos? = ¿hay perro?  $\wedge$   $\neg$  ¿hay gato?  $\wedge$  ¿hay ratón?

Definir ¿hay ratón y gato solos? =  $\neg$  ¿hay perro?  $\wedge$  ¿hay gato?  $\wedge$  ¿hay ratón?

Definir ¿hay solo dos animales? = ¿hay perro y gato solos?  $\vee$  ¿hay perro y ratón solos?  $\vee$  ¿hay ratón y gato solos?

Podemos utilizar definiciones ya realizadas para realizar nuevas. Es decir, la misma idea que con las subtareas, pero para expresiones, dejando más claro qué significa cada parte.



## Usando definiciones

Definir Alimentar animales solo sí son dos

| Sí ¿hay solo dos animales? Entonces

| | Dejar comida

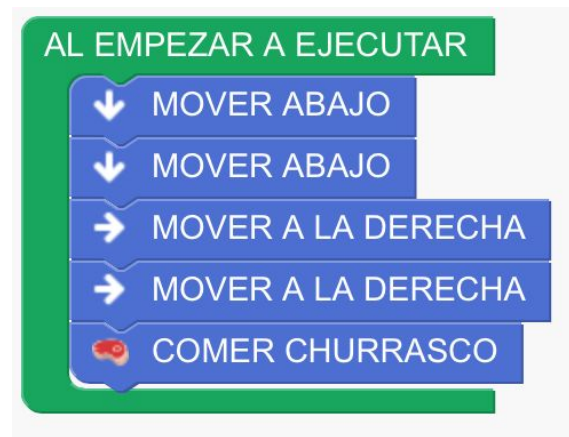
Podemos utilizar nuestras nuevas expresiones en cualquier lugar del código en dónde corresponda una expresión (dado que su uso tiene sentido para el problema, claro). Por ejemplo, en una alternativa dentro de un procedimiento.



# Momento de dudas o consultas

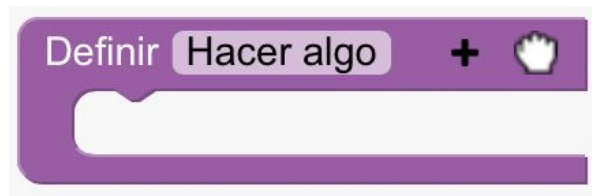
## Repaso

- Un **programa** es una **descripción** de la **solución** a un **problema computacional**.
- Un **problema computacional** es aquel que puede expresarse como una **transformación de estado**.
- En PilasBloques lo expresamos mediante bloques que se encastran entre sí, para expresar un cambio de estado en el escenario.
- Todo programa tiene un **punto de entrada**.
- Los elementos fundamentales del programa son los **comandos** (descripciones de acciones).
- Los comandos se organizan en **secuencia**, y la solución se ejecuta según esa secuencia.
- Hay **infinitos** programas que solucionan un problema. Decimos que dos programas que solucionan el mismo problema son **equivalentes**.



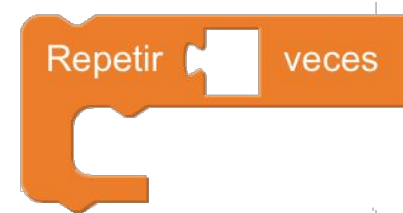
## Repaso

- Los **procedimientos** son una herramienta del lenguaje que permite definir nuevos comandos.
- Tienen un **cuerpo** y un **nombre**.
- **El nombre debe ser claro y legible, comenzar con un verbo en infinitivo y estar relacionado a su propósito.**
- La **definición** va por un lado, y el **uso (invocación o llamada)** va en el cuerpo de algún otro bloque.
- Aportan claridad, legibilidad y modificabilidad al código.
- Pueden ser **reutilizados** muchas veces.
- Permiten transmitir claramente las ideas pensadas en nuestra **estrategia**.
- Permiten separar el problema (tarea) en partes más pequeñas para su más fácil resolución (**subtareas**).



## Repaso

- La **repetición simple** es una herramienta del lenguaje que **permite cambiar el flujo** del programa (**estructura de control de flujo**).
- Permite estructurar el código de una forma distinta a la secuencia.
- Es un comando (y se puede usar junto con otros comandos en un cuerpo), pero tiene a su vez un cuerpo (es un **comando compuesto**)
- Espera una **expresión numérica** para indicar la cantidad de veces a repetir.
- **No hay que anidar repeticiones (ni ninguna otra estructura de control).**



## Repaso

- La **alternativa condicional** es una herramienta del lenguaje que **permite cambiar el flujo** del programa (**estructura de control de flujo**).
- Permite **elegir** entre posibles **ramas**, según una **condición** en el estado del programa.
- Es un comando (y se puede usar junto con otros comandos en un cuerpo), pero tiene a su vez un cuerpo (es un **comando compuesto**)
- Espera una **expresión de valor de verdad** para indicar cuándo se elige un camino y cuando otro..
- **No hay que anidar alternativas (ni ninguna otra estructura de control).**



## Repaso

- Un **comando** es la descripción de una **acción**.
- Una **expresión** en la descripción de un **dato** (un **valor, información**)
- Puede ser **numéricas**, describen una cantidad, y las usamos en la repetición simple.
- Pueden ser de **valor de verdad**, se responden con verdadero o falso y las usamos en la alternativa.
- Los **sensores** son la herramienta mediante la cual obtenemos valores de verdad basados en el estado del programa (escenario)
- Podemos usar **conectivas** para combinar sensores que describan valores de verdad, para tener valores de verdad más complejos que dependen de esos sensores.

→ Mover a la derecha

7

¿Hay banana acá?



## Repaso

- La **conjunción** une dos sensores, dando verdadero sólo cuando ambos evalúan a verdadero.
- La **disyunción** une dos sensores, dando verdadero sólo cuando alguno de los dos (o ámbos) son verdaderos.
- La **negación** aplica a un único sensor, y cambia el valor de verdad del mismo, transformando el verdadero en falso y viceversa.
- Se pueden usar expresiones con varias conectivas, y se resuelven por orden, primero las negaciones que aplican solo a un sensor, luego según indiquen los paréntesis, y si no hay más paréntesis, de izquierda a derecha.
- **Podemos definir nuestras propias expresiones en base a otras, para comunicar mejor.**

p	$\neg p$
V	F
F	V

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

## Recordatorio: Programar es comunicar!!!

- Tus programas deberían quedar claros a partir de la lectura.
- Sí leo el punto de entrada, hay un procedimiento general, en el mismo tiene que quedar más que explícita la estrategia elegida.
- Usamos procedimientos para la claridad, legibilidad y expresar la estrategia.
- Es importantísimo elegir nombres adecuados para los procedimientos que definimos.
- No anidar estructuras. Usamos procedimientos para dividir el problema en partes pequeñas y darles nombres adecuados.
- Definimos expresiones con nombres claros a partir de otras para comunicar mejor, y dividir el problema de las expresiones, también, en partes más pequeñas.

# A trabajar

Realizamos la guía práctica de actividades en papel número 3 (Práctica 3.B), todos los ejercicios.

Lo que no se termina hoy, queda para casa.





# Momento de dudas o consultas



# Definición de expresiones

Nos vemos la próxima