

NOMBRE DE LA CLASE:

# Pensamiento computacional

Duración: 45 - 60 minutos : Preparación: 10 minutos

Introducir el modelo de “Pensamiento computacional” como una forma de preparar problemas del mundo real para una representación computacional.

## RESUMEN:

Con papel y fibrones, los estudiantes deben aprender los 4 pasos del pensamiento computacional. Después de una breve introducción, deben dividirse en grupos y pensar instrucciones para dar a los otros grupos para dibujar un monstruo específico (del catálogo de monstruos preseleccionados). La tarea debe ser dividida, para que luego los equipos encuentren patrones en todos los monstruos del catálogo, y abstraigan detalles similares entre los monstruos. Deben usar esa información para crear un algoritmo (instrucciones) para que otro equipo dibuje un monstruo específico.

Los equipos deben intercambiar algoritmos y dibujar un monstruo siguiendo el algoritmo que les toque. Analizar si el dibujo coincide con lo que sus creadores pensaron.

## OBJETIVO:

Los estudiantes —

- aprenderán los cuatro pasos del pensamiento computacional

- trabajarán en grupo para resolver problemas complejos

## MATERIALES:

- Catálogo de monstruos (1 por grupo)
- Hojas de papel (3 por persona)
- Fibrones, lapiceras y lápices (1 pack por grupo)
- Tijeras

## PREPARACIÓN:

Imprimir las imágenes de los monstruos para cada grupo.

Las caras de los monstruos deben ser en papel tradicional, mientras que las otras partes deben ser impresas en papel transparente (los pueden copiar esas partes usando papel de calcar) o en papel tradicional y luego recortadas.

Entregar los materiales a cada grupo.

Escribir las palabras de vocabulario en el pizarrón o proyectarlas.

**VOCABULARIO::**

**Pensamiento computacional**—Un método de resolución de problemas que permite a los informáticos modelar problemas como soluciones digitales.

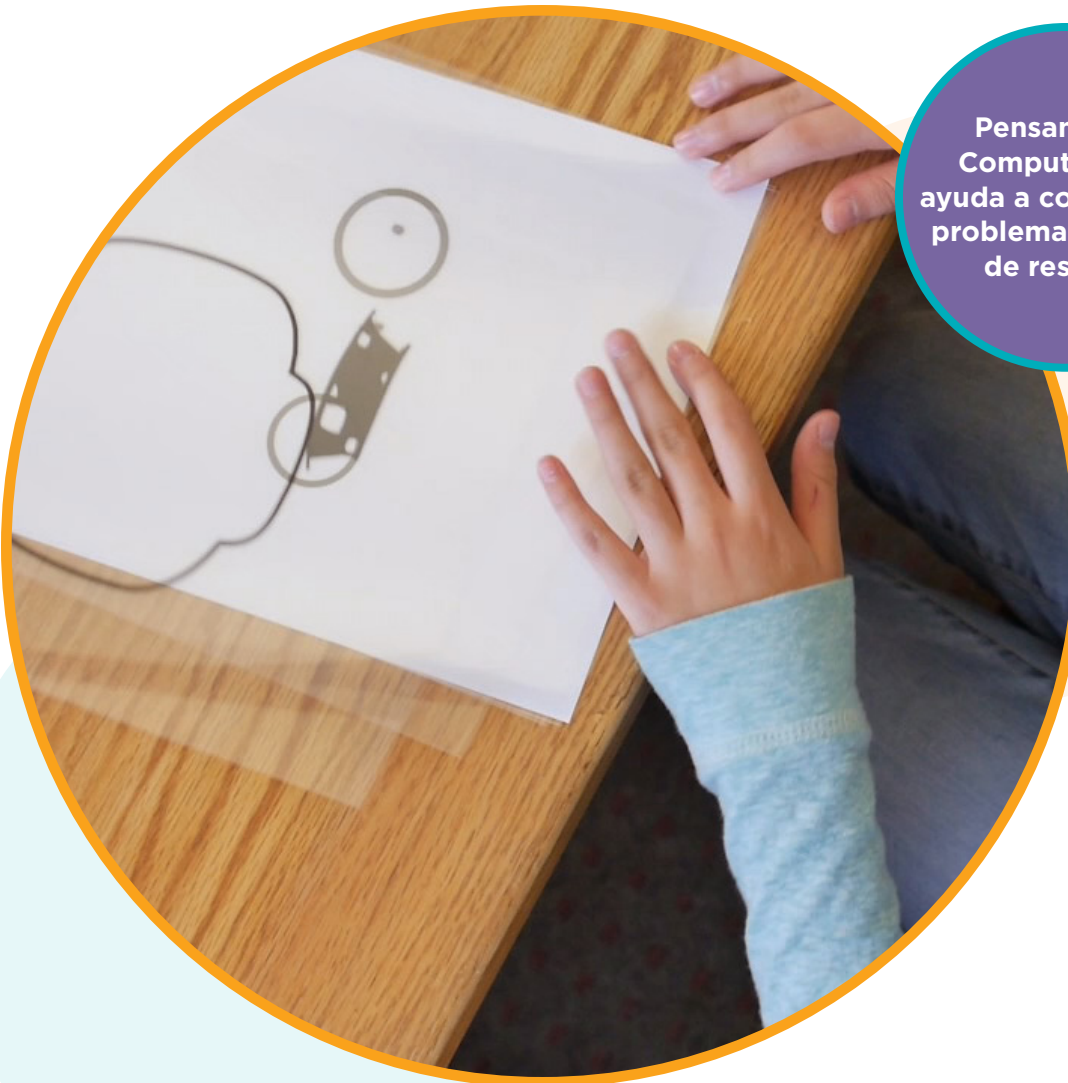
**Abstracción**—Descartar detalles para que una solución funcione para un grupo de problemas similares

**Algoritmo** —Una lista de pasos que permite realizar una tarea

**Descomponer**—Partir un problema difícil en problemas más pequeños y fáciles

**Patrón**—Algo que se repite muchas veces

**Programa**—Instrucciones que pueden ser interpretadas por una computadora



**Pensamiento Computacional ayuda a comprender problemas difíciles de resolver.**

**REPASO:**

Este repaso pretende que los estudiantes recuerden lo visto en la clase anterior. Si estás tratando de realizar las actividades sin seguir el orden, por favor incluye tu propio repaso aquí mismo.

**Preguntas para la clase:**

- ¿Qué significa “repetir” un “bloque” de código?
- ¿Cuándo usarías un bloque “if”?

**Debate:**

- ¿Puedes pensar alguna razón por la que querrías un bloque if/else, en lugar de sólo usar el bloque “if”?

**DESARROLLO:**

Hoy nos volveremos un grupo de científicos de la computación de elite. Cada uno completará una misión del mundo real, y conseguirá un “algoritmo” que puede usar para “programar” su solución. Lo siguiente es un pequeño resumen de tu misión (deberías elegir si la aceptas):

Hemos sido seleccionados para ayudar a identificar algunos monstruos del planeta Zuron. Necesitamos describir estas criaturas en base a relatos de unos testigos. Hay pocos monstruos para describir, y como puede ser desafiante te daremos algunas herramientas que van a ayudarte.

Para terminar esta operación con éxito, vamos a practicar un método llamado Pensamiento Computacional. El Pensamiento Computacional hace uso de cuatro pasos que ayudan a resolver distintas clases de problemas. Muchas personas creen que el pensamiento computacional se trata de crear soluciones que puedan ejecutarse en una computadora. Si bien eso es cierto, también ayuda a comprender problemas difíciles de resolver.

Los científicos de la computación usualmente se dan cuenta que crean programas que la gente jamás se hubiese imaginado -- cosas que nadie había creado antes. Encarar un problema que nunca fue resuelto puede ser aterrador, pero con estas simples herramientas, nada es imposible.

**Paso 1) Descomponer** —No sólo hablamos de zombies aquí. En realidad, estamos hablando de partir un gran problema en algo mucho más simple. Muchas veces, los grandes problemas consisten de muchos pequeños problemas, todos juntos en el problema mayor.

**Paso 2) Patrones**—A veces, cuando un problema tiene muchas pequeñas partes, notarás que esas partes tienen algo en común. Si no lo tienen, entonces tal vez se parezcan en mayor o menor medida a algo que ya fue resuelto con anterioridad. Si evidencias estos patrones, se vuelve más simple entender las distintas piezas que forman el problema.

**Paso 3) Abstracción**—Una vez que reconociste un patrón, puedes abstraer (ignorar) los detalles en los que difieren varias cosas distintas, y usar esto como algo general y obtener una solución que funcione para varios problemas a la vez.

**Paso 4) Algoritmos**—Cuando una solución está completa, puedes realizar una descripción que permita procesarla paso a paso, para que el resultado sea fácil de obtener.

**ACTIVIDAD: HACER UN MONSTRUO**

Cada grupo consigue el Catálogo de Monstruos. Cada monstruo necesita ser categorizado basado en la información encontrada en el catálogo. Primero, los estudiantes deben mirar a los monstruos para ver que tienen en común, y luego deben abstraer sus diferencias para crear una lista de instrucciones que permita a cualquier persona recrear el monstruo... ¡sin necesidad de saber qué monstruo están haciendo!

Hacer que una persona dibuje estos monstruos sin siquiera mirarlos pueden ser desafiante. Si miras toda la tarea puede parecer imposible. Esto es un proyecto perfecto para el pensamiento computacional.

**1) Descomponer — ¿Qué necesitamos hacer para crear un monstruo?**

Dependiendo del monstruo, puede parecer que ya está descompuesto, ¡pero lo que necesitamos es “descomponer” esta tarea! Partamos el trabajo en tareas más pequeñas. Tal vez quieras hacer esto entre toda la clase para que todos lleguen al mismo nivel.

Ejemplo:

- Ordenar los monstruos por la forma de su cara
- Mirar similitudes entre estos monstruos
- Hacer una lista de las características a identificar
- Utilizar estas características para crear un nuevo monstruo
- Describir el nuestro monstruo a tus compañeros paso a paso y que intenten rearmarlo.

**2) Patrones — ¿Qué tienen estos monstruos en común?**

¿Qué cosas poseen todos los monstruos en común? ¿Qué características son similares entre monstruos de ciertos grupos?

Ejemplo:

- Todos los monstruos tienen cabeza
- Zombus Vegitas tiene una boca de Vegitas

**3) Abstracción — ¿Qué es diferente? Extrae estas partes.**

Un monstruo puede tener ojos de Wackus, mientras que otro puede tener ojos de Spritem, pero ambos tienen ojos. Esto significa que puedes decir “Este monstruo tiene ojos de \_\_\_\_\_” y luego podremos llenar el espacio en blanco basado en qué monstruo queremos dibujar.

Ejemplo:

Crear una lista de todas las características diferentes que poseen los monstruos abstrayendo los detalles de las oraciones.

- El monstruo tiene una cabeza de \_\_\_\_\_ .
- El monstruo tiene unos ojos de \_\_\_\_\_ .
- El monstruo tiene una nariz de \_\_\_\_\_ .
- El monstruo tiene orejas de \_\_\_\_\_ .
- El monstruo tiene boca de \_\_\_\_\_ .

**4) Algoritmo — ¿Cómo expresar cada parte como una serie de instrucciones que tus compañeros pueden seguir?**

Ahora cada grupo debe ordenar sus pasos en una lista para que otros grupos puedan recrear el monstruo.

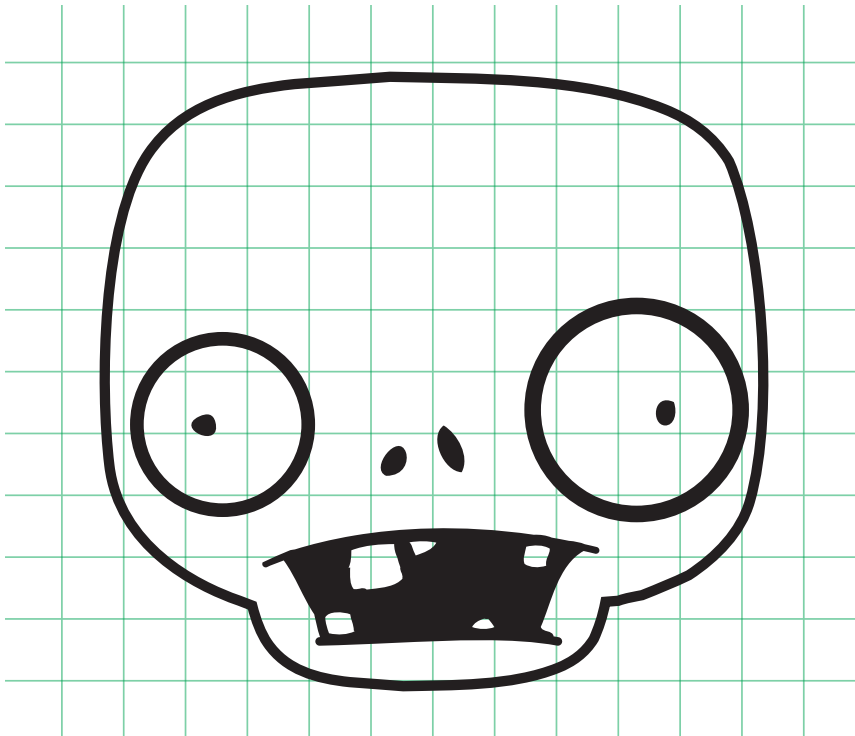
**Ejemplo:**

Crear una lista que puedas llenar.

- Dibujar una cabeza de *Wackum*.
- Dibujar ojos de *Vegetas*.
- Dibujar nariz de *Wackum*.
- Dibujar orejas de *InHideum*\*
- Dibujar boca de *Spritem*.

\*\**InHideum* significa que un monstruo carece de esta parte.

Los estudiantes deben probar su algoritmo mirando si dibuja la imagen correcta. Luego, intercambiar algoritmos con otro grupo, y dibujar el monstruo basado en ese algoritmo (sin mirar la imagen original que produce). ¿Crearon el monstruo correcto? Los estudiantes deben jugar esto una y otra vez, haciendo recrear a distintos compañeros su monstruo, o describiendo unos nuevos.



**AJUSTES:**

**Edades de 5 a 6:** Intentar hacer este ejercicio como una clase en lugar de en pequeños grupos. Usar partes recortadas (ojos, orejas, nariz) para hacer la actividad más manejable.

**Edades de 7 a 9:** Dar a los estudiantes ayuda a la hora de partir el problema y abstraer los detalles. Tal vez sea necesario hacer primero un ejemplo con ellos.

**Edades de 10 a 12:** Permitir a los estudiantes esforzarse antes de ofrecer sugerencias. Dejarlos fallar unas cuantas veces y hacer que esas equivocaciones les provean información para poder seguir avanzando. En esta edad los estudiantes se divertirán más dibujan las partes que superponiendo partes ya recortadas.

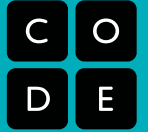
**STEPS:**

- 1) Identificar el problema.
- 2) Partir el problema en partes más pequeñas.
- 3) Abstraer los detalles para identificar patrones que se repiten en varias figuras.
- 4) Transformar a los monstruos en una lista de instrucciones para dibujarlos.
- 5) Probar las soluciones para verificar que funcionan.
- 6) Intercambiar instrucciones entre grupos y poner todas las piezas juntas como un trabajo de arte de toda la clase.

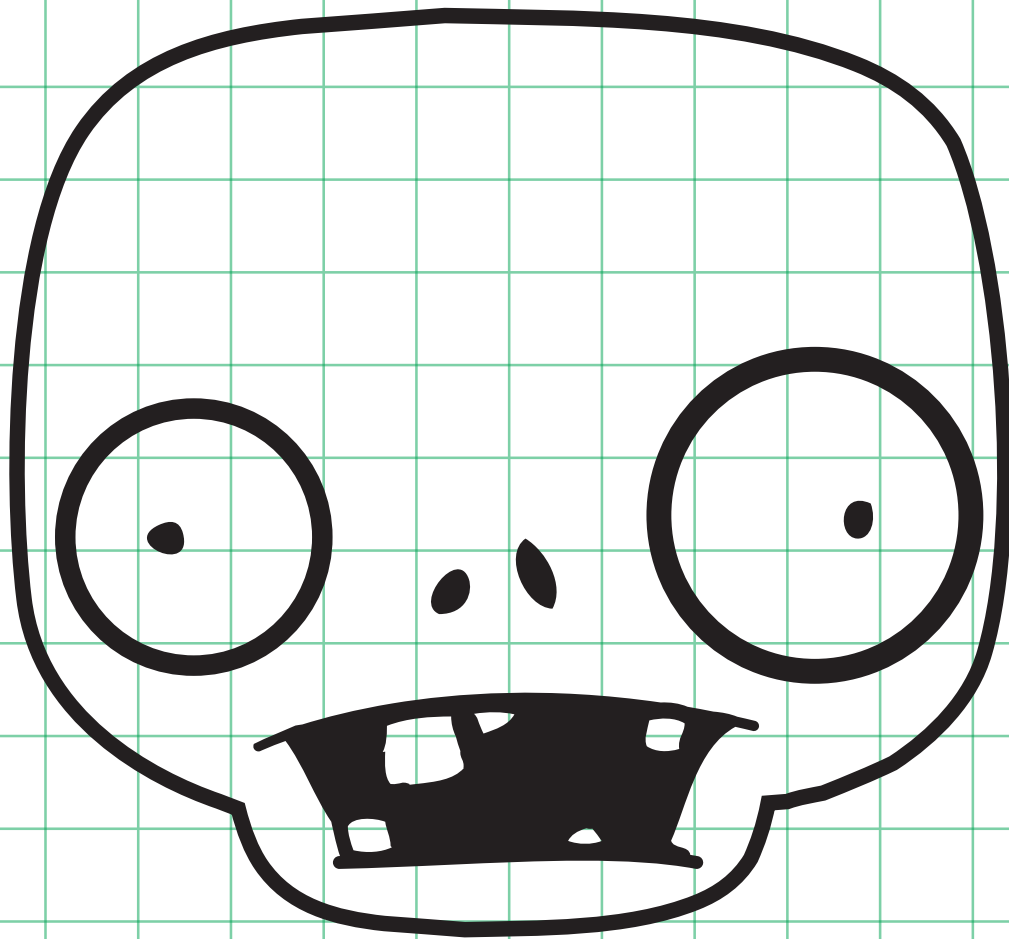
**REGLAS:**

- 1) Si el monstruo carece de una parte, la categoría de esta parte es "InHideum."
- 2) La categoría de la forma de la cabeza determina la primera mitad de la clasificación del monstruo.
- 3) La segunda mitad de la clasificación es determinada por la mayoría de las otras características.
  - Si tiene igual número de características de múltiples familias, entonces categorizar basándose en los ojos.

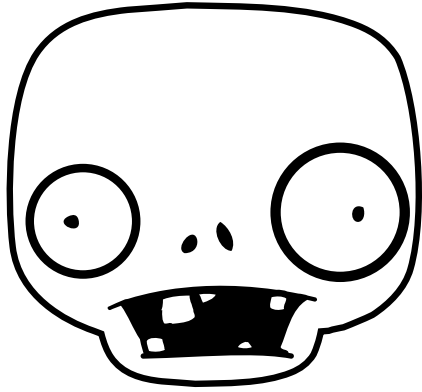
3



# Catálogo Monstruo

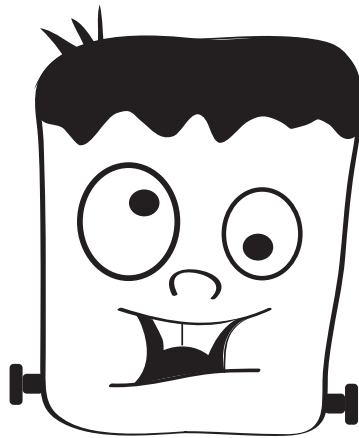


# Monsters Clasificados



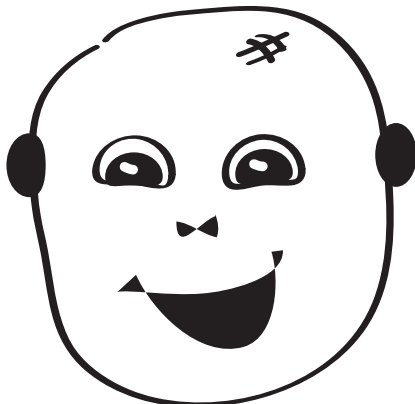
*Zombus Vegitas*

Este es el **Zombus Vegitas**, llamada así debido a su clasificación cabeza “Zombus” y su dominio de las características “Vegitas”.



*Franken Wackus*

Este es el **Franken Wackus**. Note su forma de la cabeza de “Franken” y la mayoría de las características “Wackus”.

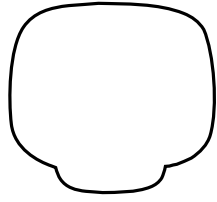


*Happy Spritem*

Este es un monstruo de la familia **Happy Spritem**. Su cabeza tiene una forma distinta “Happy” mientras que sus rasgos faciales son de la categorización “Spritem”.



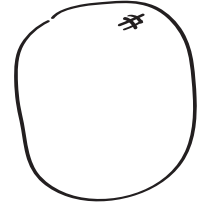
Jefes



Zombus



Franken



Happy

Ojos



Vegitas



Wackus



Spritem

Orejas



Vegitas



Wackus



Spritem

Nariz



Vegitas



Wackus



Spritem

Boca



Vegitas

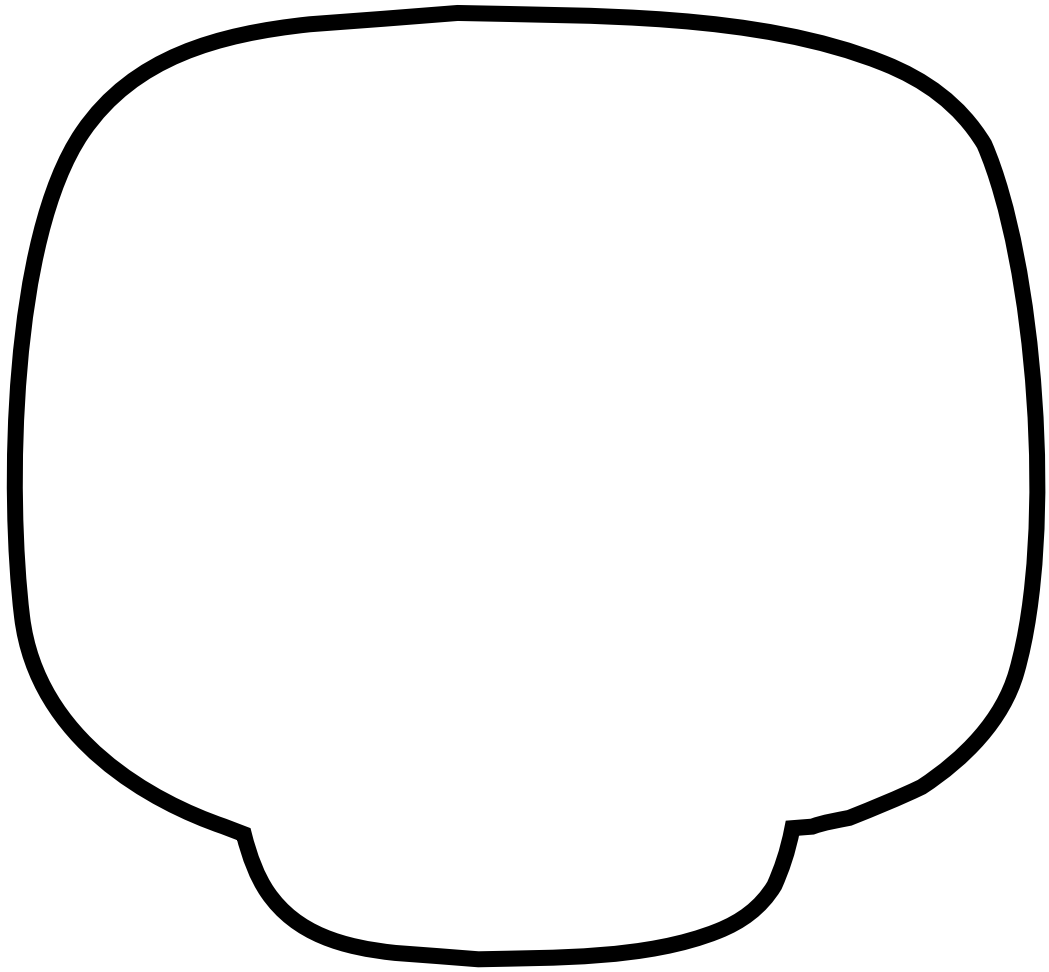



Wackus




Spritem

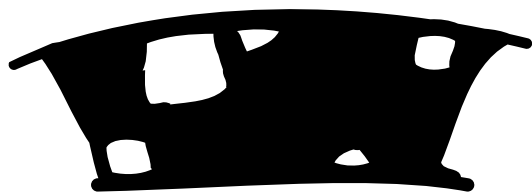

Alinear las marcas aquí



Alinear las marcas aquí



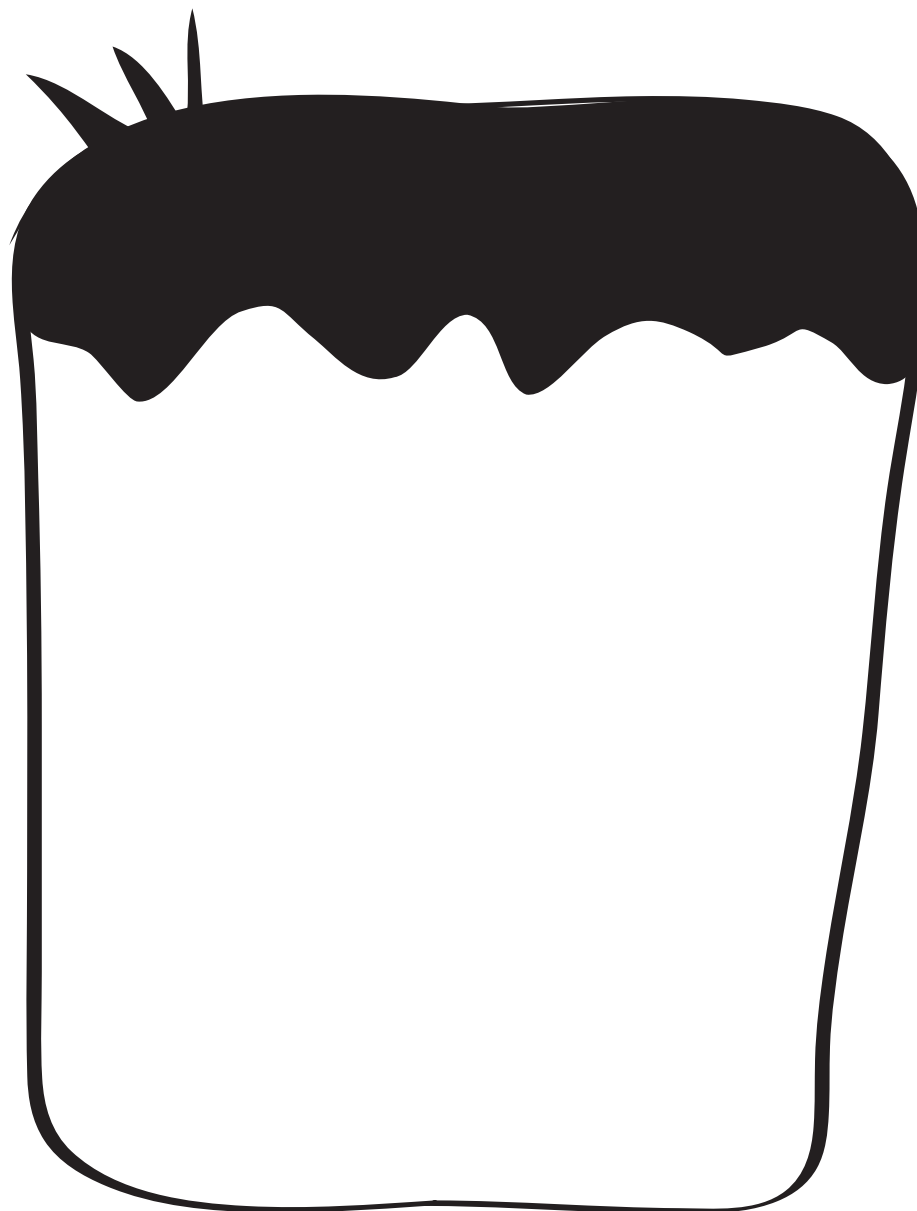
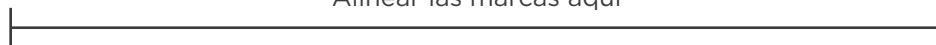
Alinear las marcas aquí




|----- Alinear las marcas aquí -----|



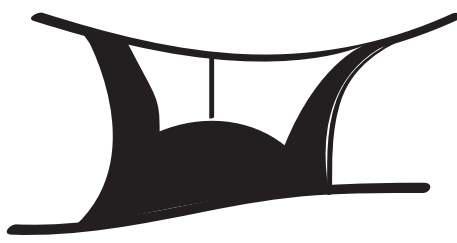

Alinear las marcas aquí



Alinear las marcas aquí

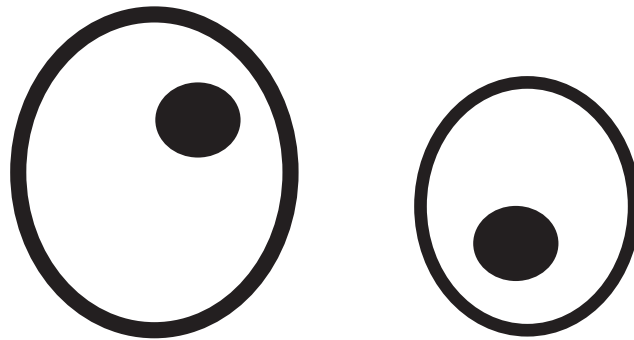



Alinear las marcas aquí






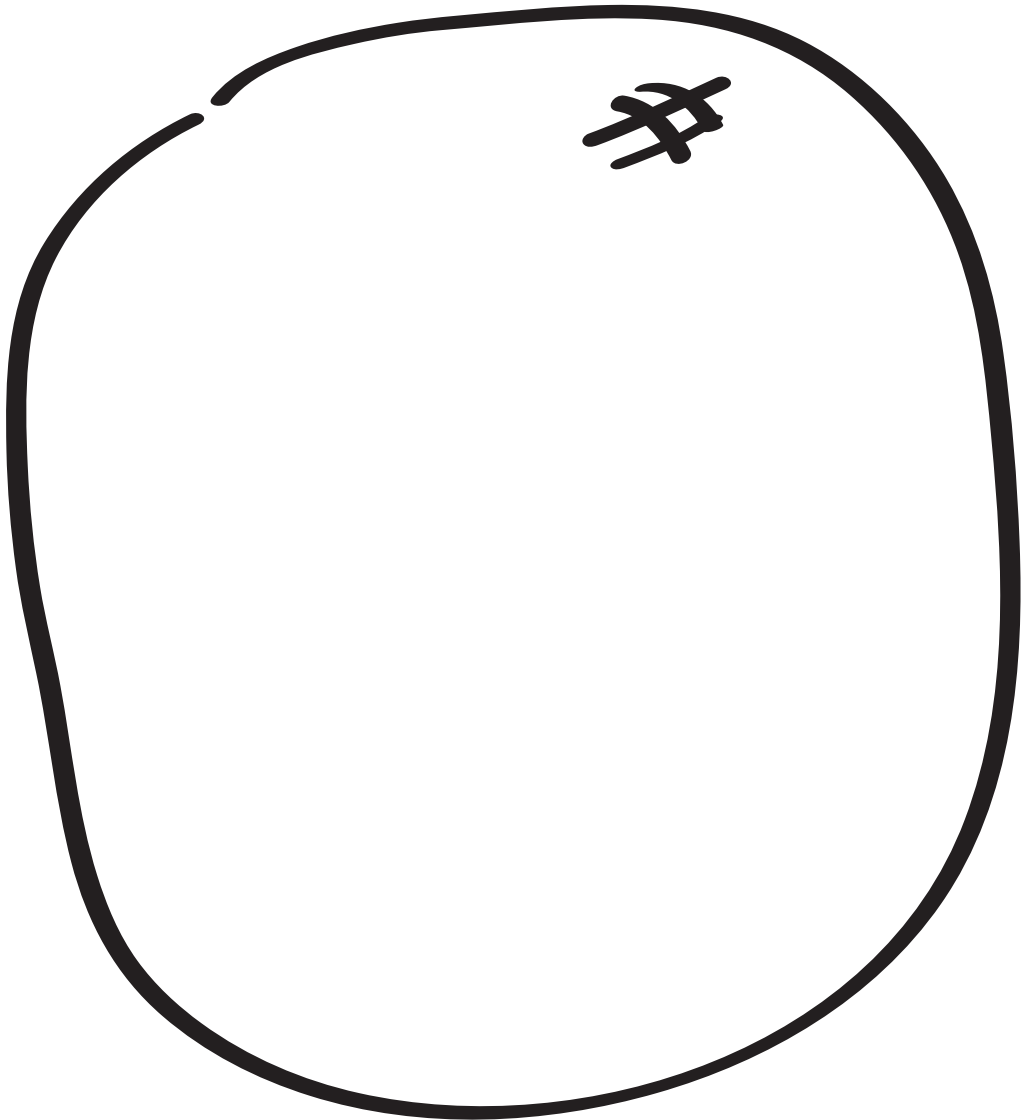
Alinear las marcas aquí



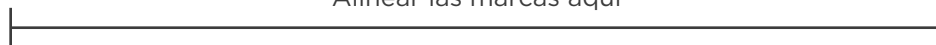
Alinear las marcas aquí




Alinear las marcas aquí




Alinear las marcas aquí



Alinear las marcas aquí



Alinear las marcas aquí



Alinear las marcas aquí

