

JADiCC 2024

PRESENTACIÓN

- **Análisis Crítico de Plataformas Educativas para Enseñar a Programar**

Jueves 27/06 18 hs



El equipo.



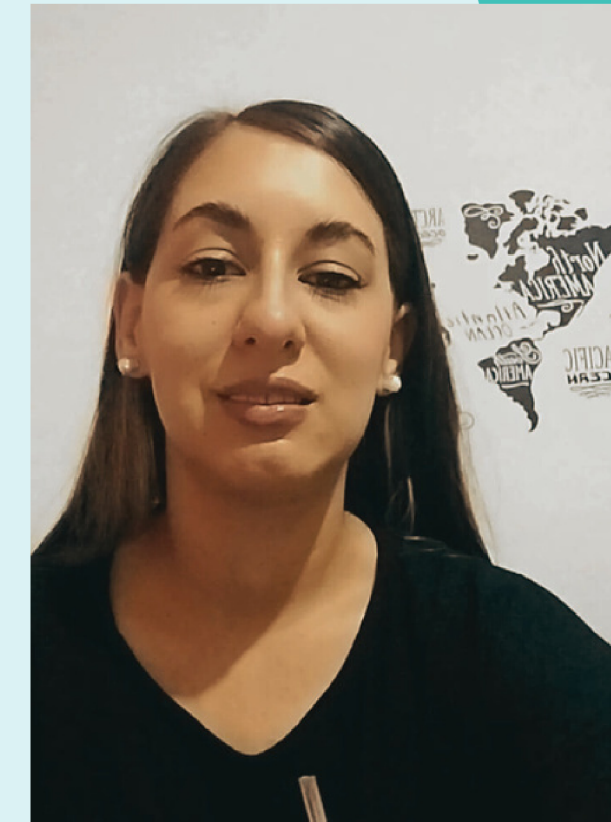
Angela Hahn

Profesora en UniPe.
Especialista en Educación y Tecnología.
Maestrando en Educación Virtual en el Nivel superior



Virginia Brassesco

Profesora en UniPe.
En proceso de doctorado en Didáctica de las Ciencias de la Computación.
Mi tema es "El razonamiento abstracto al enseñar a programar".



Yamila Ojeda

Profesora en UniPe.
Especialista Docente de Nivel Superior en Escuelas y Cultura Digital.
En proceso de obtener la Licenciatura en Tecnología Educativa.



Objetivo

Reflexionar y debatir sobre las distintas plataformas para enseñar a programar.



Introducción

Elegir una plataforma para enseñar programar **no puede depender de la moda del momento**, como docentes debemos tener claros los ejes que ponemos en juego a la hora de elegir herramientas.

El taller es un espacio para reflexionar juntos e intercambiar experiencias.

Organización del taller

01.

Presentación. ¿Quiénes somos?

Quiénes estamos en la reunión? De donde somos? En qué nivel damos clases?

02.

¿Qué plataformas usamos?

Nos dividimos en 3 grupos para trabajar qué plataformas usamos en nuestras aulas y porqué.

03.

Puesta en comun y otras posibles dimensiones de análisis

Puesta en común de los ejes propuestos en cada sala y nuevos ejes propuestos por el equipo docente.

04.

¿Cómo incorporamos estas nuevas dimensiones en las plataformas elegidas?

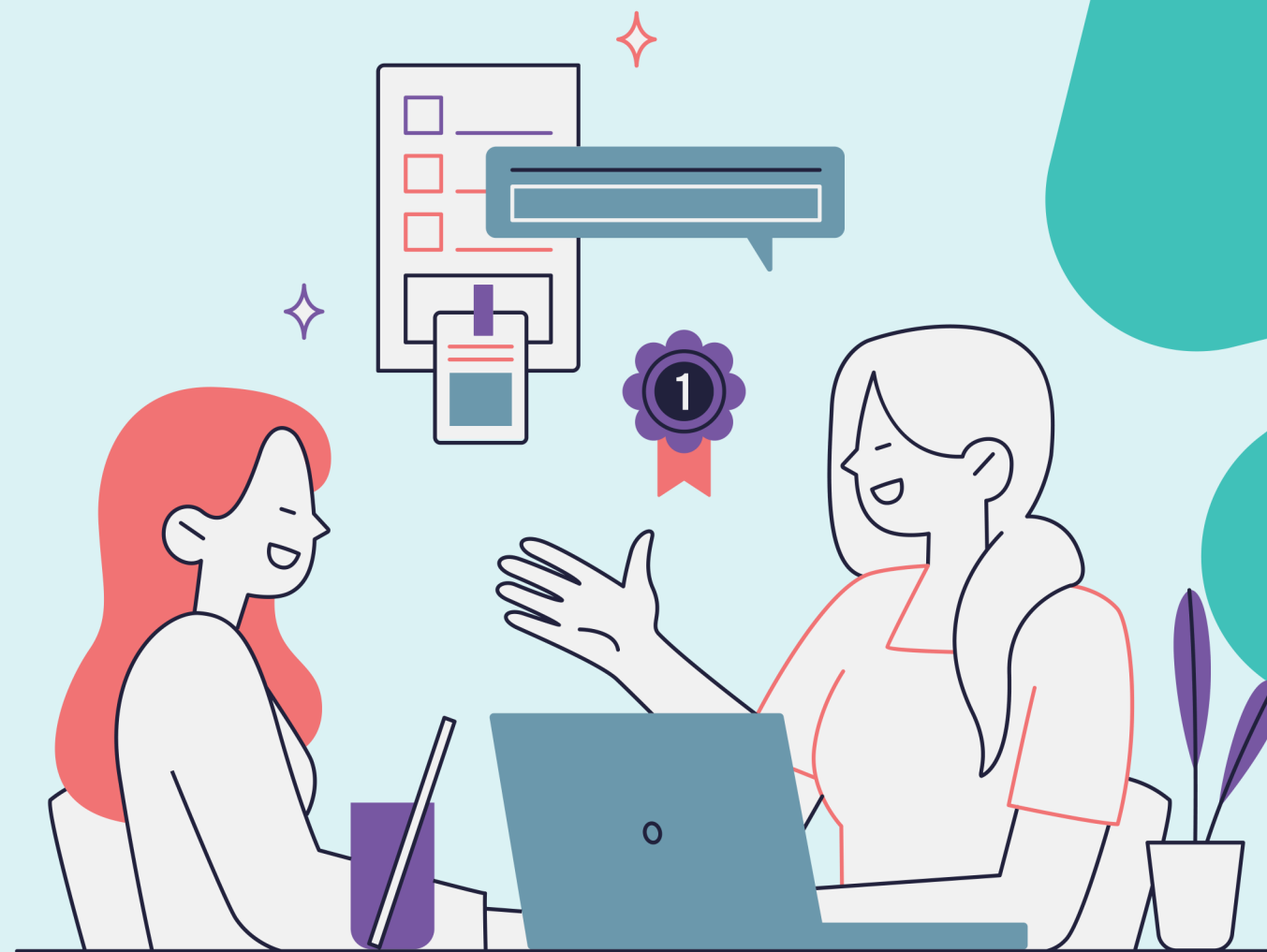
Nos separamos en 3 grupos para revisar lo que habíamos sugerido al principio.

05.

Cierre.

Cierre propuesto por el equipo docente.

***¿Quiénes
somos?***



¿En qué nivel educativo damos clases?

Inicial

1,6%

Superior

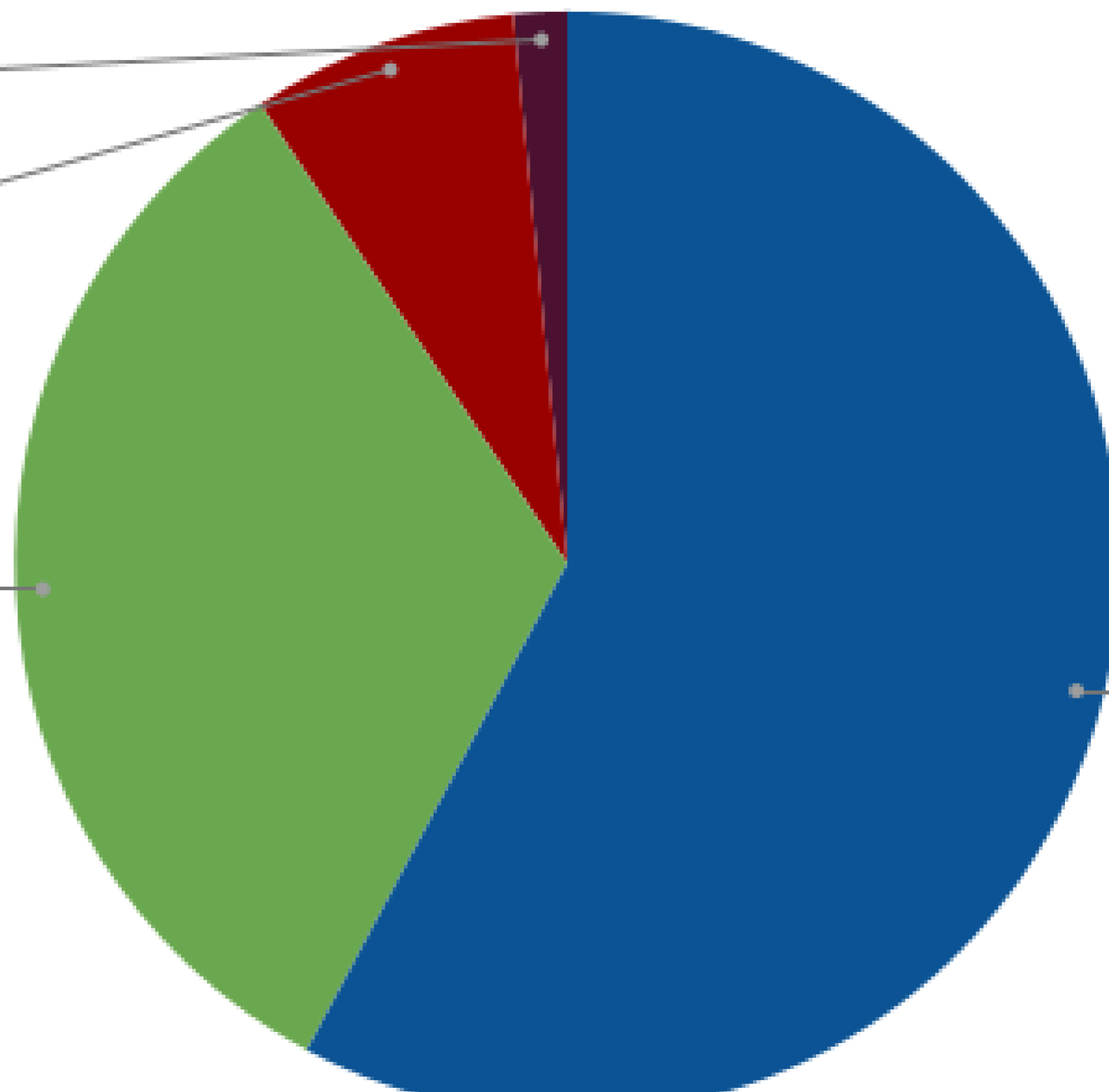
7,8%

Primaria

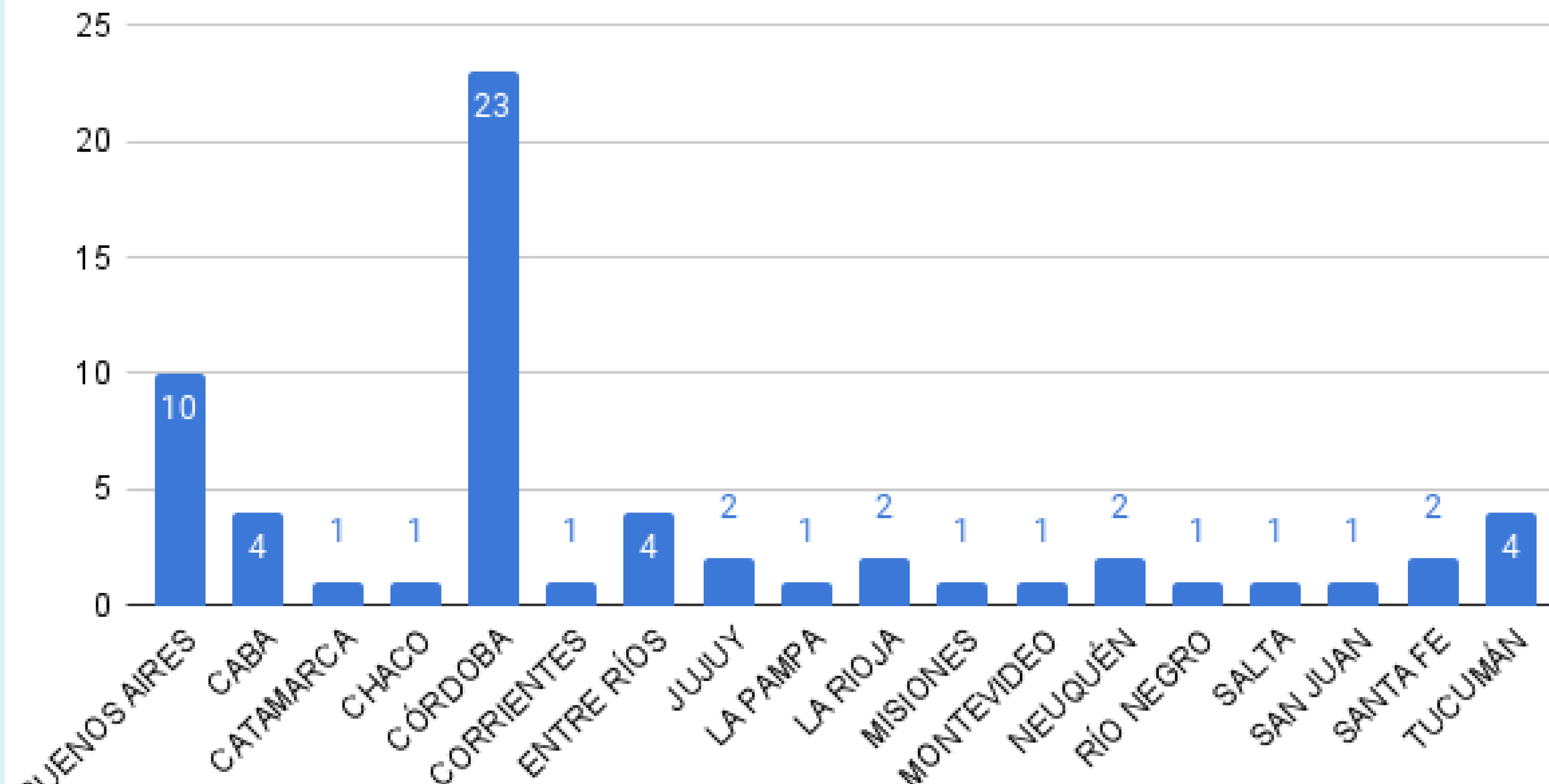
32,8%

Secundaria

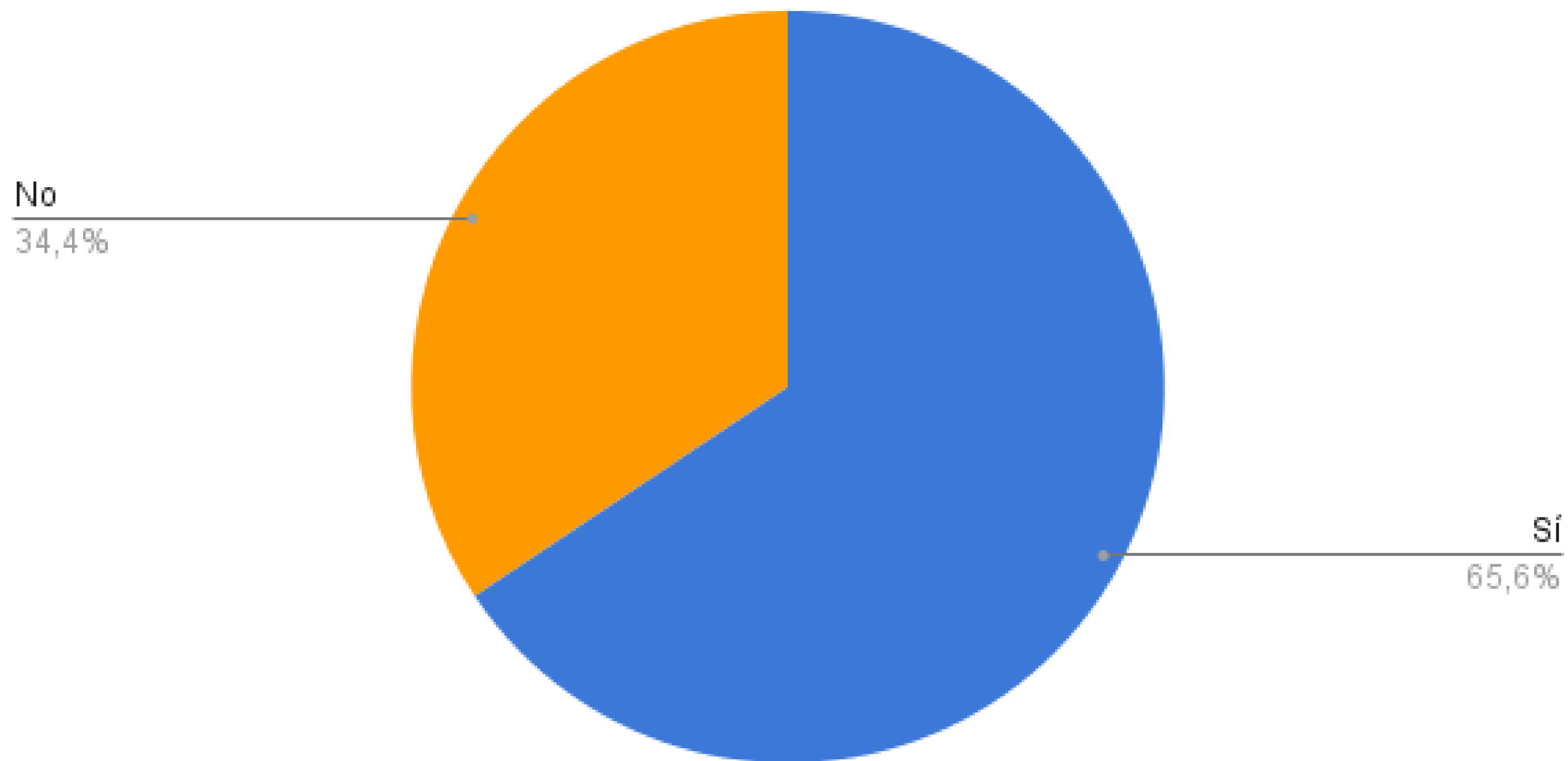
57,8%



¿En dónde damos clases?



¿Tuviste alguna formación específica de didáctica de la programación?



Nuestras motivaciones para asistir



Nos conocemos un poquito...

**Completamos el Mentimeter para
conocernos:**

- <https://www.menti.com/alamsuwa92k1>

Acceso a:

www.menti.com

Ingrese el código: 1183 8378



***¿Qué
plataformas
usamos?***



Nos separamos en 3 salas: (10 minutos)



Usaremos Mural para volcar las ideas:

<https://miro.com/app/board/uXjVK5nluqQ=/>

¿Qué plataformas usamos y conocemos?

Somos docentes de distintas provincias y realidades, ¿qué usamos en nuestras aulas?

¿Cómo usamos y por qué la usamos?

Cada persona puede contar porqué elije la plataforma que usa y por qué para poder conversar con el resto.

Volvemos a la sala principal para puesta en común

Un vocero por grupo.

¿Qué han
conversado en
cada sala?



Otras posibles dimensiones y ejes

Dimensiones basadas en el texto: JADICC 2021 ¿Scratch, Python, o qué? Criterios para elegir un entorno para enseñar a programar a principiantes. Gómez y “Fidel” Martínez López.



Algunas preguntas para empezar a pensar criterios para elegir un entorno:



¿En qué nivel de escolaridad?



¿Qué diversidades hay en mi aula?



¿Qué conceptos computacionales enseño?



Algunas facetas a tener en cuenta:

**Expresividad
y propósito
del lenguaje.**

**Forma de
construcción
de los
programas.**

**Herramientas
de soporte
pedagógico.**

Propósito general

Lenguajes de uso comercial, con propósito general:

- Python
- Java
- C++

También de propósito específico:

- latex
- gherkin
- sql
- html

Específico para enseñar a programar

- Alice
- Scratch
- App Inventor
- Gobstones
- Pilas-Bloques
- Chatbot

Ejemplo en C# de expresividad

Distintas formas de repetir

```
int[] numbers =
{ 0, 1, 2, 3, 4
};
for (int i = 0;
i < 5; i++) {

Console.Write(nu
mbers[i] + " ");
}
```

```
int[] values = {
0, 1, 2, 3, 4 };
foreach (int
value in values)
{

Console.Write(va
lue + " "); } }
```

```
int[] numbers =
{ 0, 1, 2, 3, 4
};
int i = 0; while
(i < 5) {

Console.Write(nu
mbers[i] + " ");
i++; } }
```

```
int i = 0;
do {

Console.Write(nu
mbers[i] + " ");
i++;
} while (i < 5);
```

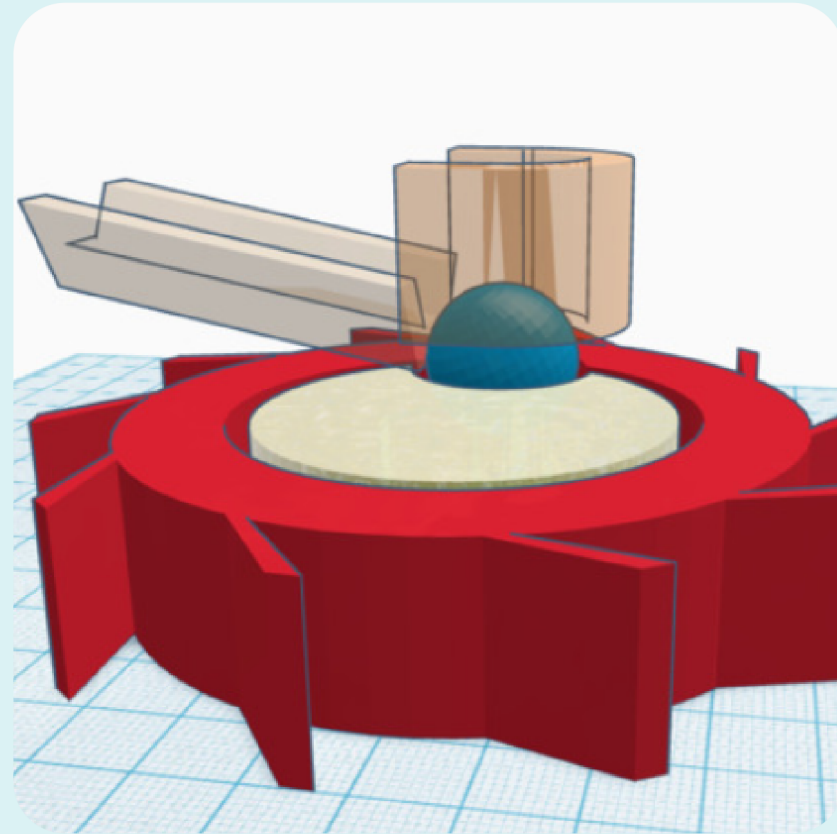
Elementos abstractos

- Arreglos
- Conjuntos
- Árboles
- Direcciones de memoria con números
- Punteros

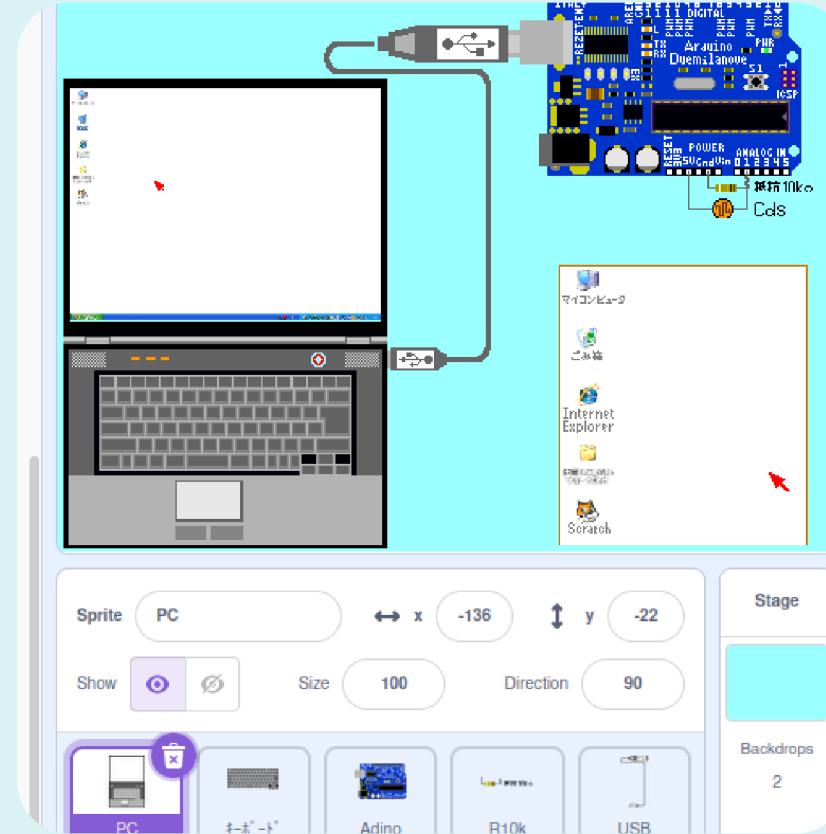
Elementos concretos

- Personajes animados
- Sensores y actuadores de un dispositivo:
 - Cámara web
 - Micrófono
 - Botones (Pulsadores)
 - Luces

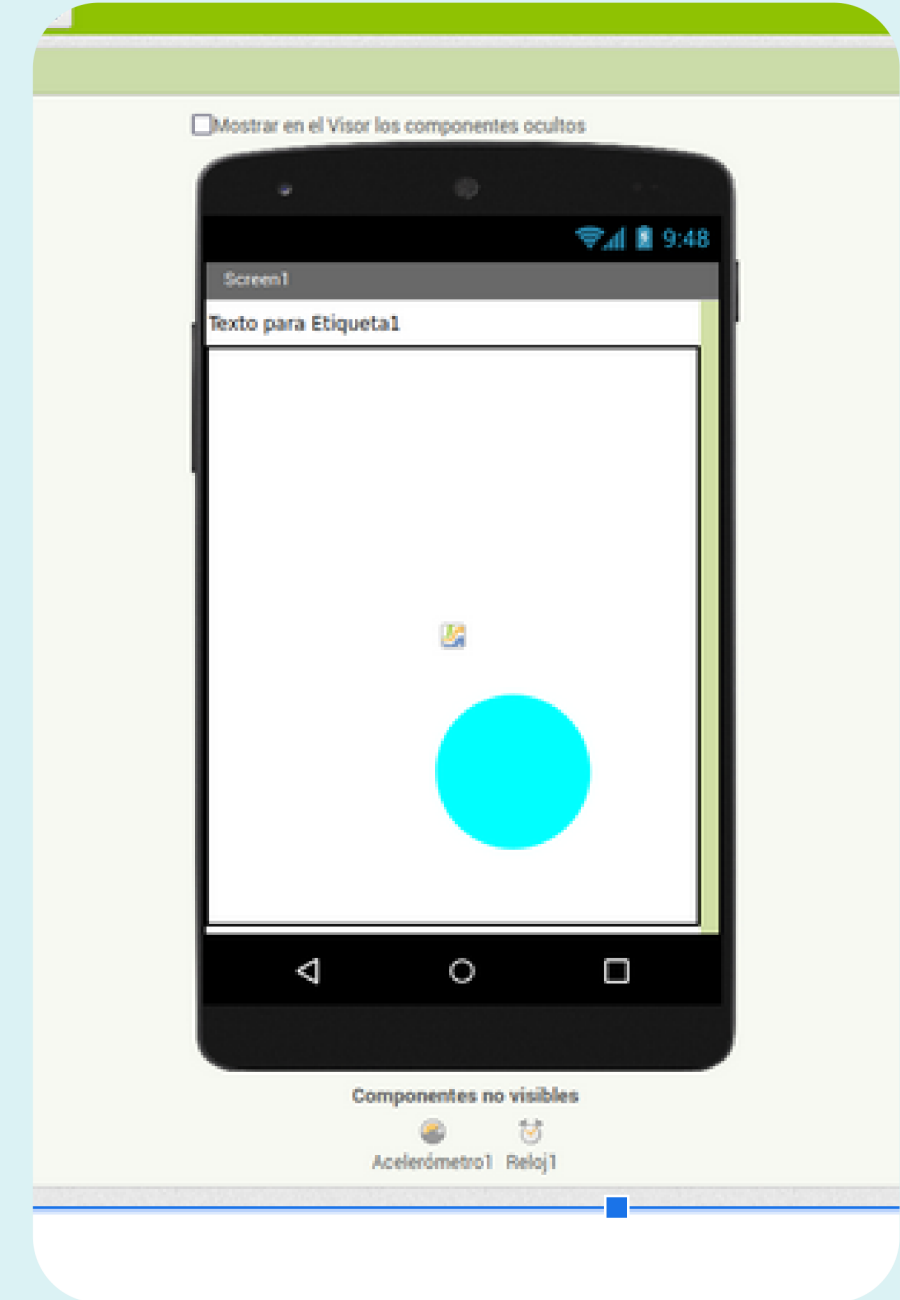
Tinkercad



Scratch + MakeyMakey +
micro:bit + Lego



AppInventor



Ejemplos de
elementos concretos

Formas de construcción

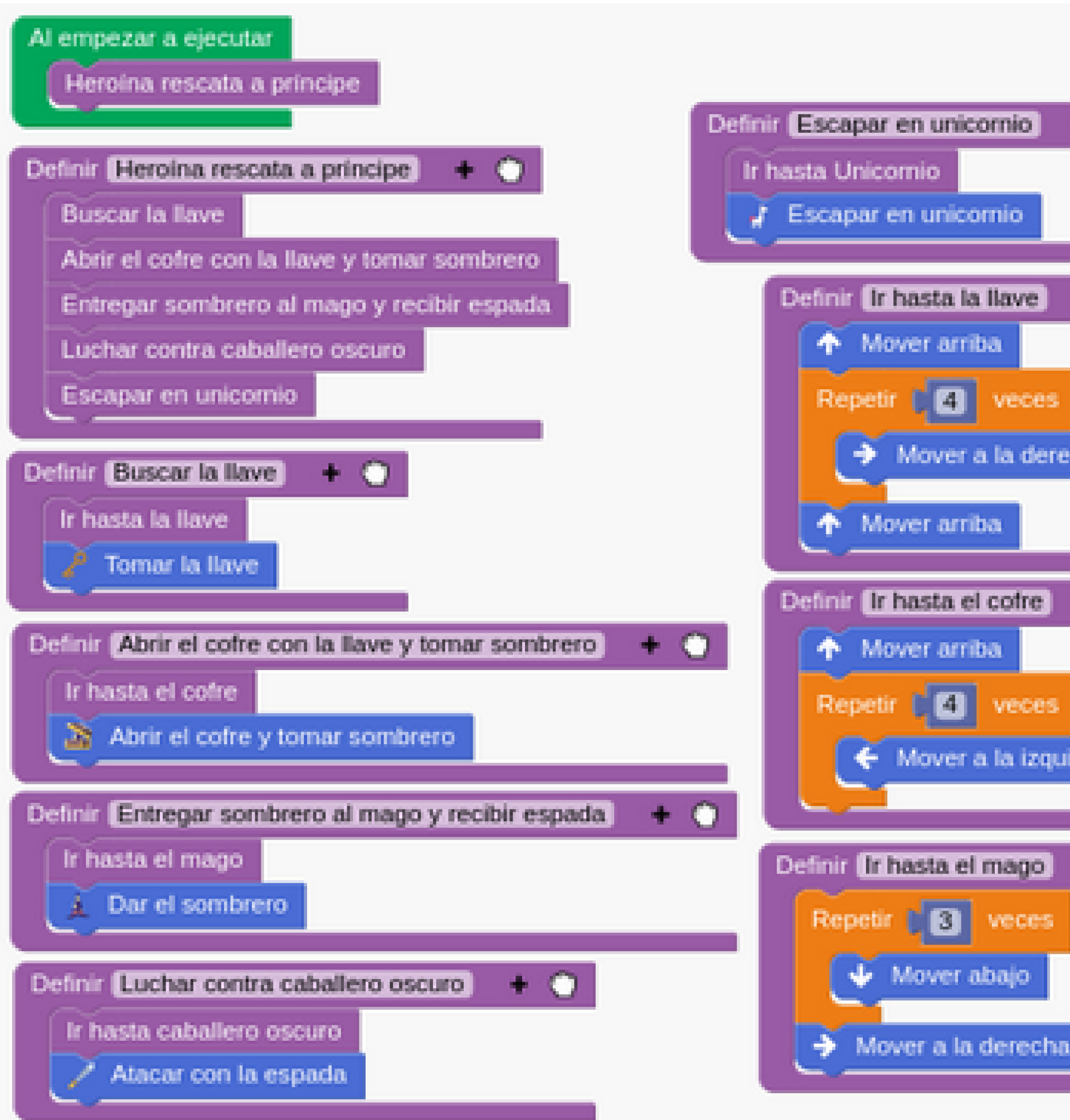
¿De qué manera el lenguaje habilita esta división en partes y ensamble?

Bloques

- Forma visual explícita de construcción
- Colores para agrupar conceptos
- Sintaxis basada en encastre
- No requiere enseñar dicha sintaxis, se puede aprender del entorno mediante indagación

Textuales

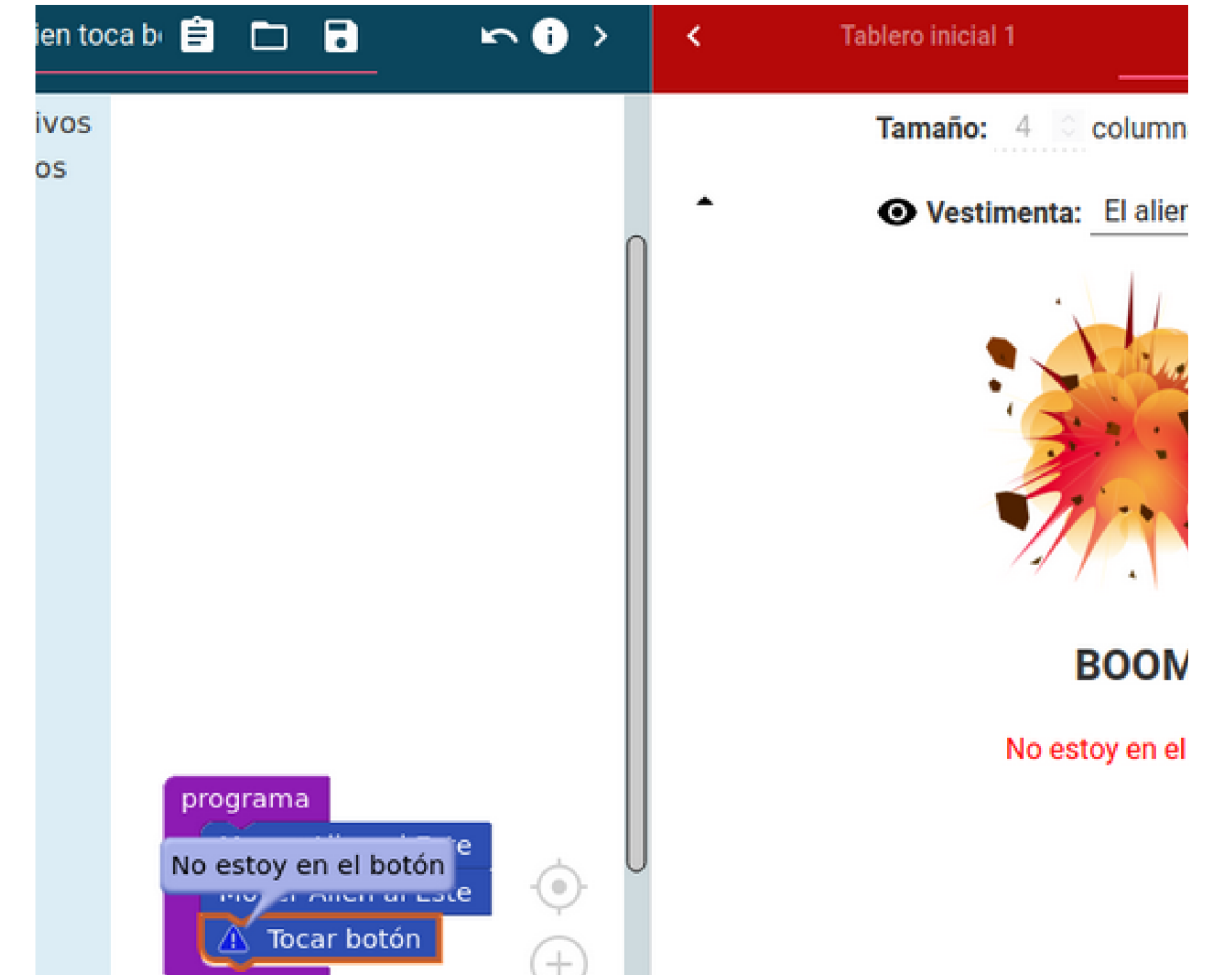
- Hay que definir la sintaxis explícitamente
- El lenguaje suele estar en inglés
- Las IDE nos pueden ayudar con errores de sintaxis pero no de semántica



Desventajas de los bloques

- Expresividad del lenguaje
- Requieren entornos digitales para programar
- El espacio de trabajo de bloques se ubica horizontal y verticalmente
 - Menos posibilidad de organización, por ejemplo indentación textual
- Combinaciones rígidas dentro de los bloques

Herramientas de soporte pedagógico



Actividades controladas

- Suelen incluir auto-evaluaciones formativas
- Mejor para grupos amplios
- Permiten trabajar conceptos específicos

Actividades Libres

- Diseñadas por un docente
- Los usuarios puedan diseñar y programar distintos tipos de aplicaciones, como animaciones, videojuegos, apps, chatbots, etc
- Desarrolla cultura Maker

Herramientas de autoevaluación

Vanilla Mode Extended Mode

 Puntuación: 13 / 21

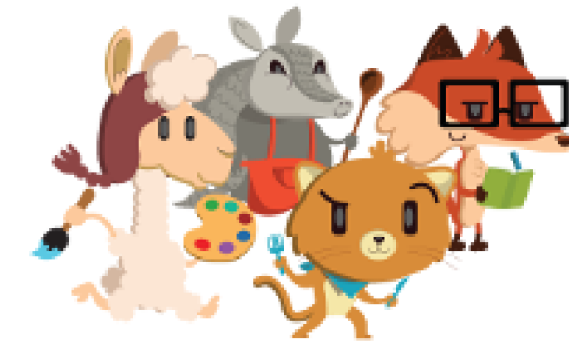
El nivel de tu proyecto es...
DEVELOPING!
!!!Eres el master del universo!!!

Malos hábitos

- 0 programas duplicados.
- 0 nombres de objetos inadecuados.
- 0 nombres de escenarios inadecuados.
- 0 código muerto.

Mejora tu nivel	Nivel
 Abstracción	1 / 3
★ Paralelismo	3 / 3
 Pensamiento lógico	0 / 3
★ Sincronización	3 / 3
 Control de flujo	2 / 3
★ Interactividad con el usuario	3 / 3
 Representación de la información	1 / 3

¡Lo lograste!



Para pensar... Tu programa no es el único que resuelve el problema. ¿Existen otras soluciones?
¿Estás usando los conceptos bien?

Actividades controladas

Problema definido, existe un oráculo para contrastar contra la solución propuesta.

Actividades Libres

- Se puede realizar detección de código duplicado y código muerto (es decir, que nunca se ejecuta). Pero, pueden ser de relevancia baja con respecto a los conceptos que se buscan impartir.
- Dr. Scratch (Moreno-León & Robles 2015) es un ejemplo de este tipo de herramientas.

Transición de Bloques a Texto

- Enfoque pedagógico a cargo del docente
- Enfoque de entornos que propicien la transición
- **Doble modalidad:** requiere mismo nivel de expresividad de lenguaje textual y de lenguaje de bloques (Alice, Pencil code)

The image shows two side-by-side screenshots of the Pencil Code web application, illustrating the transition from block-based to text-based programming.

Left Screenshot (Block-based): The interface shows the "Blocks" editor. The left sidebar contains a palette of blocks categorized by color: Move (blue), Art (purple), Text (pink), Sound (blue), Control (orange), Operators (green), Sprites (teal), and Snippets (red). The main workspace shows a script with the following blocks:
1. speed 2
2. pen red
3. for loop containing:
4. fd 100
5. rt 88
6. (empty line)

Right Screenshot (Text-based): The interface shows the "code" editor. The main workspace displays the equivalent text-based code:
1 speed 2
2 pen red
3 for [1..25]
4 fd 100
5 rt 88
6 |

¿Cómo elegir?

Entornos amigables con mis estudiantes y que me den cierto grado de control para mis actividades

Entornos amigables para el estudiante inicial, y en lo posible no abrumarlo desde el comienzo con multitud de opciones, sino que le permitan al docente ir incrementando gradualmente las opciones y herramientas disponibles.

Cierto grado de control sobre las actividades iniciales sin limitar las posibilidades de exploración de otras características.

¿Cómo elegir?

Más de un entorno y que permita indagación

Utilizar **más de un entorno**, para favorecer la centralidad de los conceptos y prácticas desarrollados, al mostrar que las ideas pueden transponerse de uno a otro con gran simplicidad y combinar dimensiones de entornos diferentes para conseguir propósitos a veces antagónicos.

Permitir tanto la **indagación** en actividades iniciales guiadas como la producción de aplicaciones propias sencillas pero interesantes.

***¿Cómo
incorporamos
estas nuevas
dimensiones en
las plataformas
elegidas?***



¡Gracias!

¿Tenes alguna pregunta?
Escríbenos.

Email

angela.hahn@unipe.edu.ar

virginia.brassesco@unipe.edu.ar

yamila.ojeda@unipe.edu.ar



<https://www.menti.com/aldix9qfuo3b>