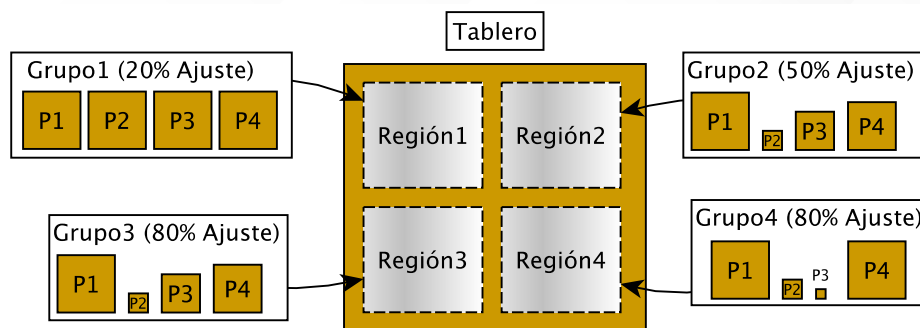


## Ciclo 1 Fundamentos de Programación

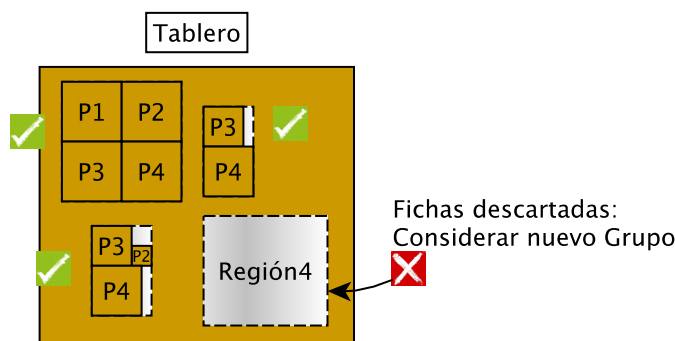
### Reto 1: División de tablero de juego



Una empresa dedicada a la construcción de juegos educativos, tiene en stock unas piezas de madera cuadradas de diferentes tamaños, las cuales sobraron de una producción previa. Estas piezas, se quieren aprovechar, y se plantea un juego para estimular competencias de espacialidad en niños, dividiendo el tablero en regiones donde máximo se puedan colocar 4 de estas piezas, pero con un nivel de dificultad intermedio, por lo tanto las regiones deben ser bastante ajustadas. Proponer las regiones que luego se distribuirán en tableros que se ordenarán a un proveedor, sería una tarea muy extensa, por lo tanto, se propone un criterio que se implementará en un software para automatizar el proceso, el cual se ilustra en las Figuras 1 y 2, donde se muestra cómo se definen las regiones a partir de grupos de 4 fichas seleccionadas aleatoriamente del stock.



**Figura 1.** Tablero con regiones sin definir para las piezas agrupadas.



**Figura 2.** Regiones generadas después de realizar el proceso

La estrategia para formar las regiones ajustadas, consiste en que por lo menos 2 de las 4 piezas cumplan con el criterio de ajuste, el cual variará para generar regiones con diferentes dificultades para que el niño ubique las piezas. Dicho porcentaje de ajuste, podría establecerse entre 0 y 1, donde 0 obligaría a que las piezas coincidan exactamente con el área promedio, 0.5 permitiría una diferencia de hasta la mitad del área promedio y 1, una diferencia de hasta la totalidad del área promedio. En la Ecuación 1 se especifica este criterio:

$$DiferenciaPermitida = PorcentajeAjuste * AreaPromedio$$

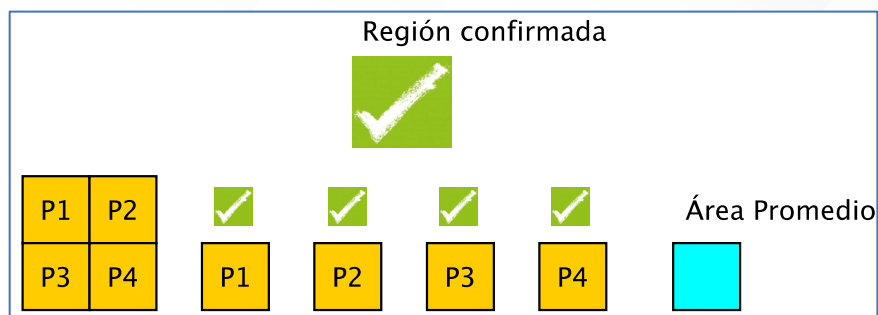
**Ecuación 1.** Diferencia permitida entre el área de una pieza y el área promedio del grupo

Como las piezas pueden presentar un área mayor o menor al área promedio, entonces, la diferencia de cada pieza se debe expresar aplicando valor absoluto, como se indica en la Ecuación 2:

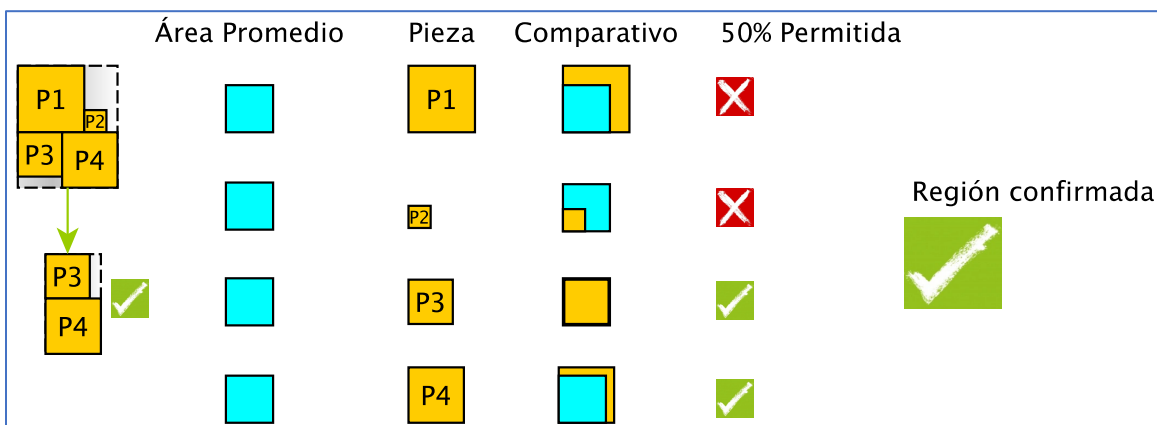
$$|AreaPromedio - AreaPieza|$$

**Ecuación 2.** Medida de ajuste para el bloque de 4 piezas

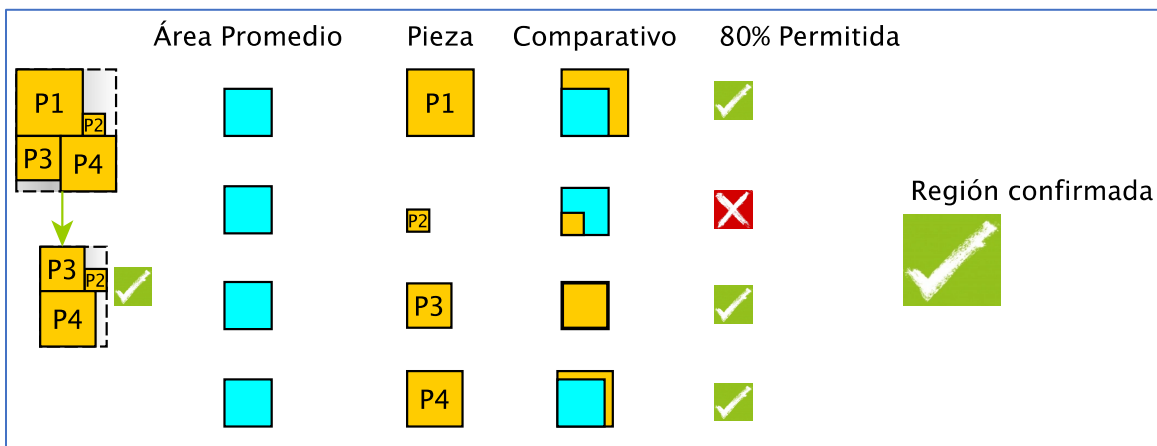
En las Figuras 3, 4, 5 y 6 se presenta gráficamente la aplicación del criterio:



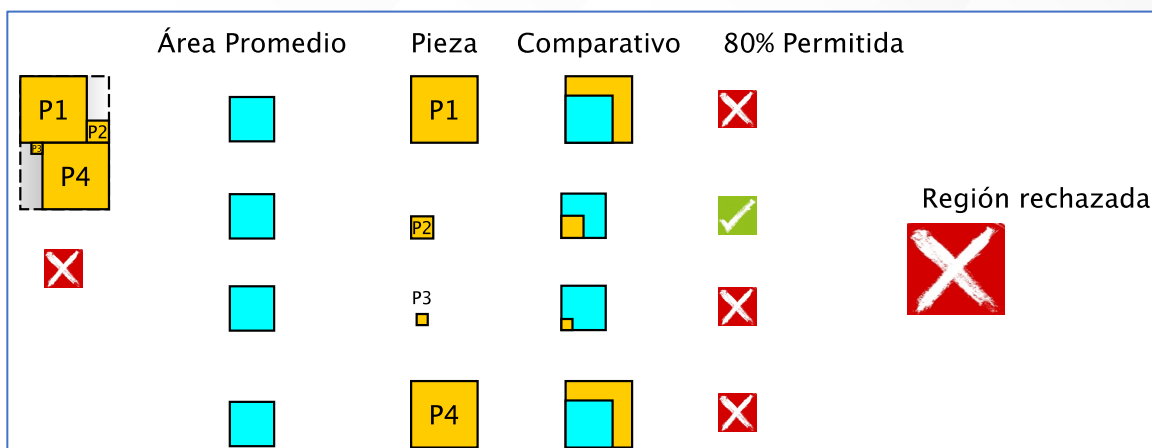
**Figura 3.** Generación de la Región 1 del Tablero.



**Figura 4.** Generación de la Región 2 del Tablero.



**Figura 5.** Generación de la Región 3 del Tablero



**Figura 5.** Generación de la Región 4 del Tablero

Nótese en la Figura 4, que con un porcentaje de ajuste de 0.5, es decir, 50%, solamente las piezas 3 y 4 aprobaron el criterio de ajuste, porque **el valor absoluto de la diferencia del área que ocupa cada una, y el área promedio, es menor o igual que la diferencia permitida**. La región resultante es mostrada entonces en el sector izquierdo, descartando a P1 y P2 en el empaquetado. El mismo caso, en la Figura 5, muestra la incidencia de aumentar el porcentaje de ajuste, donde otra pieza más aprobaría el criterio.

Se necesita entonces implementar una función (algoritmo), que reciba el lado de cada una de estas piezas cuadradas en pulgadas, junto con el porcentaje de ajuste, y determine si puede generarse una región ajustada que incluya parcial o totalmente las piezas del grupo, informando el número de piezas que contendrá la región, y el área acumulada que será ocupada por estas. Si la región no incluye por lo menos dos piezas, informar la cantidad de piezas que fueron rechazadas, y reportar la necesidad de una nueva agrupación de piezas del stock. En la siguiente tabla se muestran las regiones/grupos de piezas de las figuras, especificando cómo deberían ser procesadas (entradas y reporte de salida) en la implementación:

| Entradas |       |       |       |                  | Salida  |
|----------|-------|-------|-------|------------------|---|
| Lado1    | Lado2 | Lado3 | Lado4 | PorcentajeAjuste | return  |
| 3        | 3     | 3     | 3     | 0.2              | Región confirmada: piezas aprobadas = 4, área piezas ubicadas = 36    |
| 3        | 1     | 2     | 2.5   | 0.5              | Región confirmada: piezas aprobadas = 2, área piezas ubicadas = 10.25 |
| 3        | 1     | 2     | 2.5   | 0.8              | Región confirmada: piezas aprobadas = 3, área piezas ubicadas = 19.25 |
| 3        | 1     | 0.5   | 3     | 0.8              | Fichas descartadas: 3, considerar nuevo grupo de piezas del stock     |

A continuación, se presenta la especificación de los tipos de datos de la función, y la estructura que debe tener la función que se va a desarrollar:



### Entradas

| Nombre           | Tipo de Dato | Descripción  |
|------------------|--------------|--|
| lado1            | float        | Valor flotante mayor a 0 y menor o igual a diez (pulgadas de la pieza) |
| lado2            | float        | Valor flotante mayor a 0 y menor o igual a diez (pulgadas de la pieza) |
| lado3            | float        | Valor flotante mayor a 0 y menor o igual a diez (pulgadas de la pieza) |
| lado4            | float        | Valor flotante mayor a 0 y menor o igual a diez (pulgadas de la pieza) |
| porcentajeAjuste | float        | Valor flotante (decimal) entre 0 y 1                                   |

### Salida

| Tipo de Retorno | Descripción   |
|-----------------|---|
| str             | "Fichas descartadas: {piezasDescartadas}, considerar nuevo grupo de piezas del stock"               |
| str             | "Embalaje confirmado: piezas aprobadas = {piezasAprobadas}, superficie ocupada = {areaEmpaquetado}" |

### Prototipo de la Función

```
#Tipado de la función
def generadorRegiones(largo1: float, largo2: float, largo3: float, largo4: float, porcentajeAjuste: float) -> str:
    #Escribir aquí la solución
    pass
```

**Nota:** En la plataforma debe subirse una función con **el mismo nombre, la misma cantidad de argumentos, y el retorno debe tener exactamente la estructura de la salida que se presenta en este documento.**