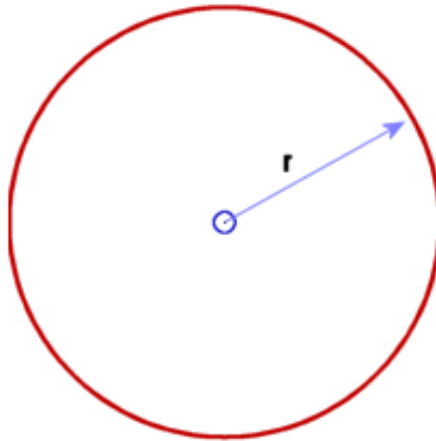


## CUADRADO DE SIERPINSKI. PROGRAMACIÓN EN PHP PDF

### PROCESO DE CONSTRUCCIÓN

#### **ITERACIÓN 0**

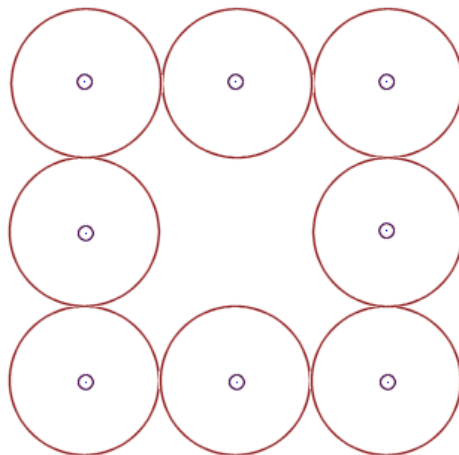
Partimos de un círculo de radio  $r$ .



En esta iteración, que llamamos 0 por ser el punto de inicio, tendremos un número de círculos  $n_0 = 1$

#### **ITERACIÓN 1**

Con 8 círculos iguales, los montamos formando un cuadrado:

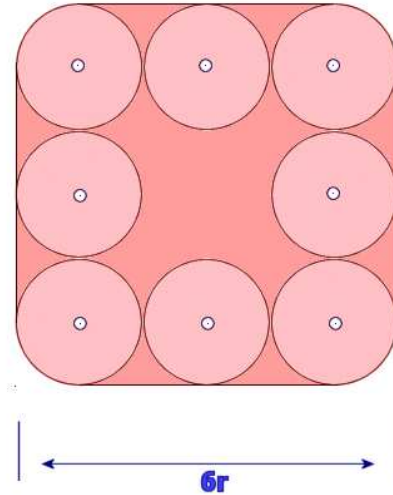


En esta iteración, tendremos un número de círculos  $n_1 = 8$

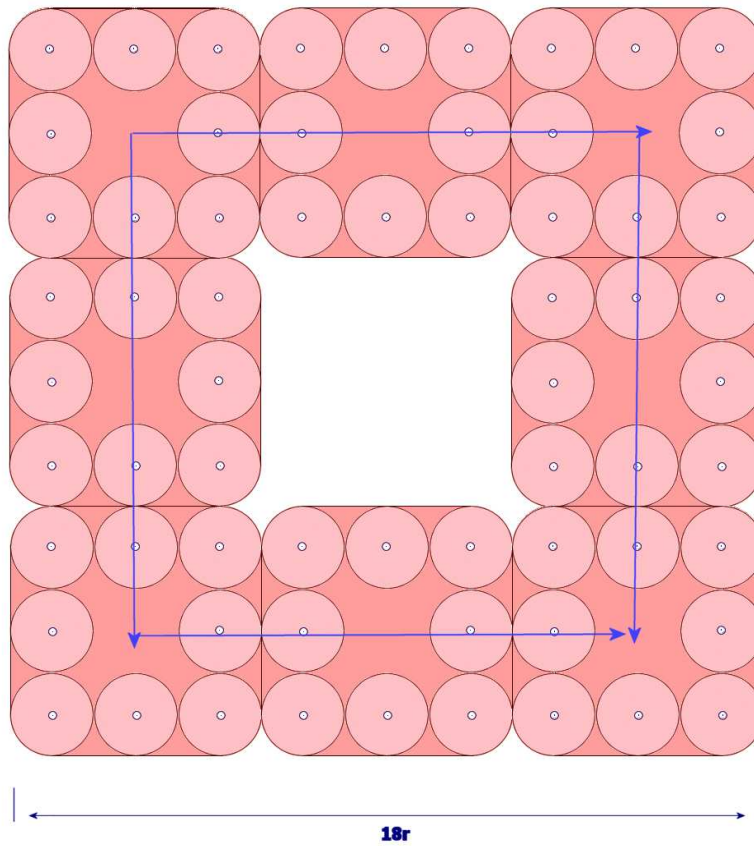


## ITERACIÓN K

Conforme vamos aumentando las iteraciones, observamos que toda la estructura de la iteración anterior se repite 8 veces de modo similar. Si consideramos una iteración como una estructura encerrada en un "cuadrado":

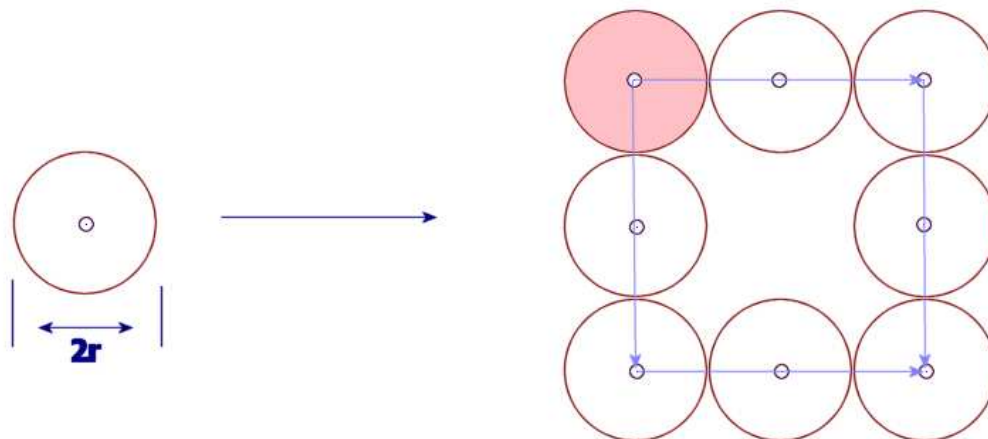


Bastaría con repetir esta estructura ocho veces. A cada estructura la llamaremos BLOQUE:



Para ello, vamos a controlar los desplazamientos de cada bloque, tanto en horizontal (eje de abscisas) como en vertical (eje de ordenadas), por lo que deberemos fijar 7 movimientos distintos (ocho dejando el primer bloque invariante)

En la primera iteración:



Así pues, partiremos de un punto cualquiera  $(x_0, y_0)$  sobre el que construir la iteración

El centro del círculo superior lo situamos sobre dicho punto, por lo que con respecto a este punto no sufre desplazamiento alguno:

$$v_0 = (0,0)$$

Desplazándonos hacia la derecha y hacia abajo, los vectores de traslación son:

$v_0 = (0,0)$	$v_1 = (2r,0)$	$v_2 = (4r,0)$
$v_3 = (0,2r)$		$v_4 = (4r,2r)$
$v_5 = (0,4r)$	$v_6 = (2r,4r)$	$v_7 = (4r,4r)$

En la segunda iteración la tabla de traslaciones es

$v_0 = (0,0)$	$v_1 = (6r,0)$	$v_2 = (12r,0)$
$v_3 = (0,6r)$		$v_4 = (12r,6r)$
$v_5 = (0,12r)$	$v_6 = (6r,12r)$	$v_7 = (12r,12r)$

En las iteraciones sucesivas, es fácil deducir que:

- El número de círculos es  $C_k = 8^k$
- El ancho del bloque generado es  $d_k = 3^k \cdot 2 \cdot r$

- La tabla de desplazamientos es:

$v_0 = (0,0)$	$v_1 = (3^{k-1} \cdot 2r, 0)$	$v_2 = (3^{k-1} \cdot 4r, 0)$
$v_3 = (0, 3^{k-1} \cdot 2r)$		$v_4 = (3^{k-1} \cdot 4r, 3^{k-1} \cdot 2r)$
$v_5 = (0, 3^{k-1} \cdot 4r)$	$v_6 = (3^{k-1} \cdot 2r, 3^{k-1} \cdot 4r)$	$v_7 = (3^{k-1} \cdot 4r, 3^{k-1} \cdot 4r)$

## **TRATAMIENTO DE LOS BLOQUES POR CADA ITERACIÓN**

Cada bloque estará formado por dos matrices de datos, correspondientes a las abscisas y ordenadas de los centros de cada círculo.

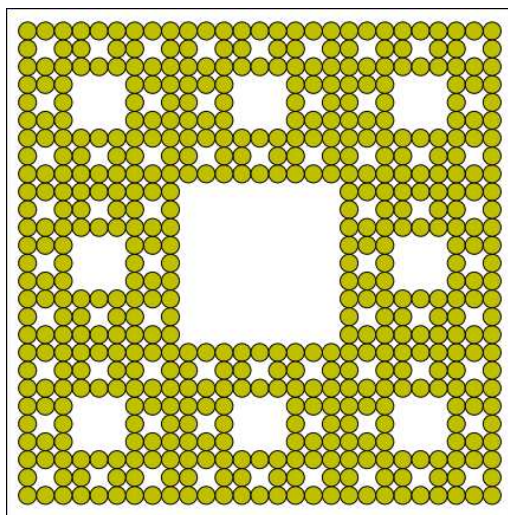
Para construir una nueva iteración ampliamos las matrices 8 veces, una por cada vector de traslación

Para formar las iteraciones en modo BUCLE, al finalizar cada iteración, seguiremos la secuencia de pasos siguientes:

- 1º.- Borrar las tablas  $X_0$  e  $Y_0$
- 2º.- Crear nuevas tablas  $X_0$  e  $Y_0$  vacías
- 3º.- Copiar las tablas  $X_1$  e  $Y_1$  en  $X_0$  e  $Y_0$  respectivamente
- 4º.- Borrar las tablas  $X_1$  e  $Y_1$

Para que todo esto funciones correctamente, es necesario establecer el punto de inicio donde las tablas  $X_0$  e  $Y_0$  son de dimensión 1 y contienen las coordenadas del punto cartesiano a partir del cual comenzaremos a crear el triángulo,  $x_0$  e  $y_0$  respectivamente.

Así, si seguimos hasta la iteración 3º, obtendremos:

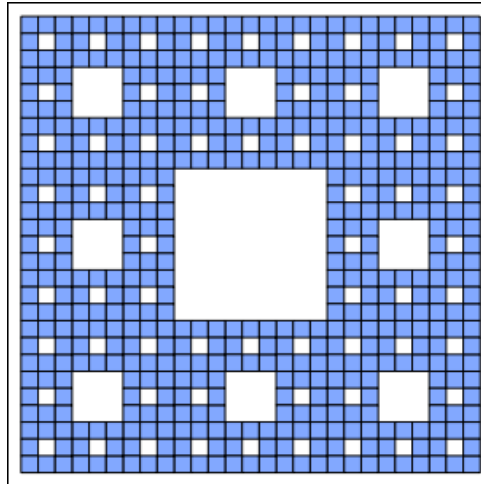


## **FIGURA BASE**

Todo este proceso se ha realizado tomando como figura base un círculo de radio  $r$ .



Si en lugar de un círculo, utilizamos un cuadrado obtendríamos cuadrados de Sierpinski semejantes:



## **CODIFICACIÓN A PDF**

No hay problemas importantes a la hora de codificar en PHP-PDF salvo que el sistema de coordenadas tiene el origen en el vértice superior izquierdo de la hoja; el eje X aumenta hacia la derecha, pero el eje Y funciona al revés, aumenta hacia abajo, por lo que los dibujos saldrán invertidos verticalmente salvo que corriamos esta diferencia.