

# Introducción a Ciencias de la Computación

## Semestre 2020-1

### Práctica 10

Profesor: Amparo López Gaona  
Ayud. Lab.: Marco Iván Olea Olvera

Jueves 03 de Octubre de 2019

## 1 Objetivos

- Practicar el uso de la estructura de control `for`.
- Familiarizarse con la construcción de interfaces gráficas mediante lenguajes de programación.

## 2 Introducción



*Processing* es una biblioteca gráfica y un lenguaje de programación para aprender a programar en el contexto de las artes visuales.

Un programa escrito en Processing se llama *sketch*, y la sintaxis de este lenguaje está muy influenciada por Java. Processing incluye un Entorno de Desarrollo Integrado (o *IDE* por sus siglas en inglés) donde se pueden escribir y ejecutar *sketches* fácilmente. Pero en realidad, código escrito en Processing es traducido a Java nativo antes de ser ejecutado, y posteriormente ese código es compilado a lenguaje de la máquina virtual de Java (JVM). De hecho, un sketch no es más que una subclase de una clase especial de Processing llamada **PApplet**.

Como este es un curso en el que están aprendiendo Java, no vamos a utilizar el lenguaje de processing ni mucho menos el IDE. Escribiremos programas directamente en Java que utilicen clases ofrecidas por la biblioteca de Processing, las cuales están contenidas en el archivo `core.jar`.

## 3 Descripción

Si queremos construir una aplicación de ajedrez, primero que nada debemos hacer que de alguna manera se dibuje un tablero en la pantalla, y eso es lo único que contempla esta práctica. Debes escribir un programa de Java que muestre un tablero de ajedrez en la pantalla, utilizando la biblioteca de Processing.

La estructura básica de un programa de este tipo es una subclase de la clase `processing.core.PApplet` que implemente/sobreescriba los siguientes métodos:

- Un método `main` el cual lo único que hace es invocar el método `PApplet.main(String)`, que recibe como parámetro el nombre completo de una subclase de `PApplet`.
- El método `public void settings()` que sólo sirve para fijar las dimensiones de la ventana que se creará, y esto lo hace a través del método `size(int, int)`.

- El método `public void setup()`. Este se invoca una sola vez al iniciar el programa, y sirve para inicializar variables.
- El método `public void draw()`. Este se ejecutará continuamente `frameRate` veces por segundo hasta terminar el programa, donde el valor de la variable `frameRate` depende de la computadora en que se ejecute el programa.

Puedes consultar la documentación de PApplet y el resto de las clases de `processing.core` aquí <http://processing.github.io/processing-javadocs/core/processing/core/PApplet.html>.

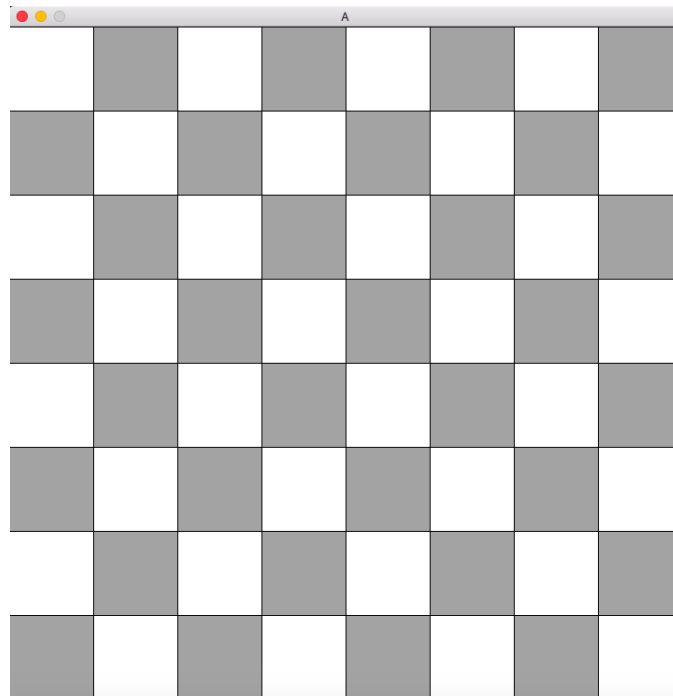
## 4 Desarrollo

1. Crea una subclase de PApplet. El nombre de esta clase y del paquete en el que está contenido pueden ser lo que tú quieras.
2. Escribe el método `main()` como se requiere para iniciar el programa.
3. Sobreescribe los métodos `settings()`, `setup()`, y `draw()`. En `settings()` únicamente se fijará el tamaño de la ventana; `setup()` estará vacío (aún así lo tienes que escribir); en `draw()` estarán las instrucciones necesarias para dibujar el tablero de ajedrez, los únicos métodos de PApplet que necesitas son `fill(int, int, int)` y `rect(float, float, float, float)`.

Punto extra si haces que el tamaño del tablero esté en función del largo y/o ancho de la pantalla de la computadora en que se ejecute el programa.

Notas:

- No está permitido usar el paquete default (tus clases deben estar dentro de un paquete).
- Todos los métodos que sobrescriban un método definido en su superclase deben tener la etiqueta `@Override`.
- Recuerda documentar tu código y seguir las convenciones y reglas de estilo de Java.



## 5 Entrega

En esta única ocasión se te pedirá subir código compilado a GitHub.

<https://github.com/usuario123/ICC-2020-1>

```
├── ...
├── practica10
│   ├── src
│   │   └── <paq>
│   │       └── <Nom>.java
│   ├── lib
│   │   └── core.jar
│   ├── build
│   │   ├── <paq>
│   │   └── <Nom>.class
└── README.md
```

Donde <Nom> es el nombre de tu clase y <paq> es el nombre del paquete donde está contenida dicha clase.