Universidad Zaragoza
1542

# Informática
# Prácticas de Laboratorio

Grado en Estudios en Arquitectura

**Curso 2018-2019**

Escuela de Ingeniería y Arquitectura
Departamento de Informática e Ingeniería de Sistemas
Área de Lenguajes y Sistemas Informáticos
Febrero de 2019

# Práctica 1:

# Introducción a Processing

## Objetivo de la práctica

Los objetivos de esta primera práctica de la asignatura son los siguientes:

- Familiarización con el computador y los elementos básicos de su sistema operativo.
- Familiarización con el entorno de desarrollo de programas en Processing.

## 1. Processing

### 1.1. Introducción

Processing es un lenguaje de programación sencillo (relativamente) y muy didáctico, al permitir trabajar de un modo cómodo con aspectos como elementos y, por tanto, poder observar de un modo visual el resultado de nuestros programas. Como veremos en sesiones de prácticas posteriores, esto no es así en la mayoría de los lenguajes de programación. Por ejemplo, en el lenguaje de programación Java lo normal es comunicarse con los programas usando texto, siendo la entrada un texto introducido por el teclado y mostrándose como salida un texto por la consola de salida.

El lenguaje fue propuesto en 2001 por científicos del Instituto Tecnólogico de Massachussets (MIT). Está diseñado como una capa adicional sobre Java, por lo que es un lenguaje de programación completo (como Java) con funcionalidad añadida para facilitar el uso de gráficos, animaciones, sonidos, etc.

Processing se distribuye junto a un entorno de desarrollo integrado que se distribuye de forma libre y gratuita. Este entorno se encuentra ya instalado en las aulas de prácticas, pero el Apéndice 1 explica su instalación.

Este guión está basado en el libro “Daniel Shiffman. Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction. Morgan Kaufmann, 2014”.

La página web oficial http://www.processing.org contiene un manual de referencia (sección *Reference*) con documentación sobre todos los elementos de Processing, siendo especialmente útil para encontrar información sobre el funcionamiento y los parámetros de los métodos.

### 1.2. El entorno de desarrollo

El entorno de desarrollo de Processing tiene un diseño minimalista que facilita su uso. Tal y como se aprecia en la Figura 1, la interfaz gráfica posee las siguientes partes:

a) un menú de opciones: fichero, edición, esbozo, herramientas y ayuda.

b) una lista de iconos para acceder a algunas opciones habituales: ejecutar, detener una ejecución, crear un nuevo fichero, abrir un fichero existente, salvar el fichero actual y exportar la aplicación.

c) un editor de texto (la zona central con fondo blanco) que permite editar el código fuente de varios ficheros, cada uno de los cuales se abrirá en una nueva pestaña,

d) una zona inferior donde se muestran mensajes al usuario, y

e) el número de línea del código actual, en la esquina inferior izquierda.


*Fig. 1. Interfaz gráfica del entorno de desarrollo de Processing*

Como veremos más adelante, al ejecutar un programa se abrirá una nueva ventana donde se mostrará el resultado de la ejecución.

# 2. Dibujando figuras geométricas

Los píxeles son puntos de la pantalla del ordenador. Nuestro primer objetivo será dibujar una línea recta en la pantalla. Para ello, necesitamos especificar al ordenador dónde queremos dibujarla. Aunque hay varias formas de determinar la ecuación de una recta, lo más cómodo es probablemente definir dos puntos por los que pasa.

A modo de ejemplo, vamos a dibujar el segmento comprendido entre los puntos (10, 0) y (40, 50). Para ello, escribimos el siguiente código en la pantalla de Processing y ejecutamos pulsando el icono de *Run*:

```
line(10, 0, 40, 50);
```

¿Es éste el resultado que esperabas? Es muy probable que no sea así, por lo que conviene hacer una aclaración importante: Processing fija el origen del sistema de coordenadas de la pantalla en la esquina superior izquierda de la ventana de ejecución, si bien el origen del sistema de coordenadas cartesianas suele representarse en la esquina inferior izquierda.

Ya has escrito tu primera línea de código. Básicamente, se está proporcionando una *orden* al ordenador (dibujar una línea), junto a una serie de *argumentos* que indican cómo dibujar la línea (las coordenadas de los dos puntos que delimitan el segmento). En esta asignatura veremos que ese tipo de órdenes se conoce como *método*. Si pensamos en la línea de código como una frase, el método es el verbo y los argumentos los objetos.

Prueba a modificar la línea anterior de la siguiente manera:

```
Line(10, 0, 40, 50);
```

¿Sabrías decir qué está sucediendo? Processing es sensible a (distingue entre el uso de) las mayúsculas y minúsculas, lo que se conoce en inglés como *case-sensitive*.

Recuerda utilizar la opción de salvar cada vez que quieras guardar una copia del código.

Ahora prueba a dibujar líneas en diferentes puntos de la pantalla, comprobando si los resultados obtenidos coinciden con los esperados.

A continuación, vamos a intentar dibujar otras figuras geométricas. ¿Qué parámetros crees que se necesitan en cada uno de los siguientes casos?

a) Punto
b) Triángulo
c) Cuadrilátero
d) Rectángulo
e) Elipse

Seguramente has observado que en alguno de estos casos hay más de una solución posible. ¿Podrías poner algún ejemplo?

Para dibujar las anteriores figuras geométricas, existen los siguientes métodos en Processing:

a) `point` recibe como parámetros las 2 coordenadas x e y del punto.
b) `triangle` recibe 6 parámetros: las coordenadas x e y de los 3 vértices del triángulo.
c) `quad` recibe 8 parámetros: las coordenadas x e y de los 4 vértices del cuadrilátero.
d) `rect` recibe 4 parámetros: las coordenadas x e y del vértice superior izquierdo, la anchura y la altura del rectángulo.
e) `ellipse` recibe 4 parámetros, los correspondientes al rectángulo que constituye la *bounding box* (el rectángulo mínimo que contiene por completo a la elipse).

En el caso del rectángulo y la elipse, es posible que los 4 parámetros sean interpretados de una manera diferente: las coordenadas x e y del punto central del rectángulo, la anchura y la altura. Para que esto se haga así, es necesario usar previamente la orden

```
rectMode(CENTER);
```

Para que vuelvan a considerar la interpretación vista anteriormente, basta usar la orden

```
rectMode(CORNER);
```

En el caso de las eclipses, habría que utilizar las órdenes:

```
ellipseMode(CORNER);
```

y

```
ellipseMode(CENTER);
```

respectivamente. Por defecto, Processing considera los valores `rectMode(CORNER)` y `ellipseMode(CENTER)`.

Intenta dibujar un círculo centrado en (20, 30) con radio 10 usando los dos modos anteriores.

Ahora escribe lar instrucciones necesarias para obtener por pantalla el contenido de la Figura 2, teniendo en cuenta que puede haber más de una solución posible.

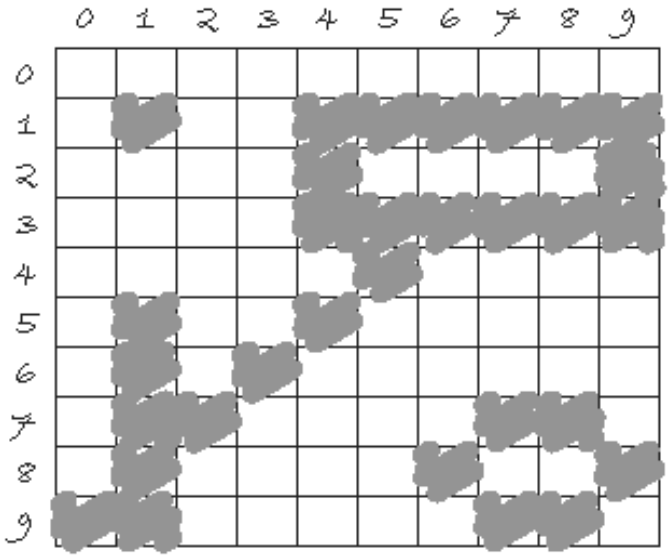


*Fig. 2. Ejemplo de salida por pantalla*

# 3. Añadiendo color

Hasta ahora no nos hemos preocupado de un elemento importante al trabajar con información gráfica: el color. Para comenzar, consideraremos el caso más simple de las imágenes, el blanco y negro. Los colores en la gama de grises se pueden representar mediante un número entero en el rango [0, 255], donde 0 corresponde al negro, 255 al blanco y los números intermedios corresponden a los valores intermedios, ordenados en orden decreciente de tonalidad, como se ilustra en la Figura 3.

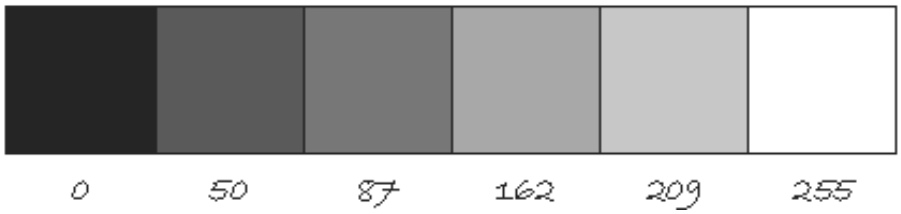


*Fig. 3. Ejemplos de colores en blanco y negro*

En Processing, las figuras tienen un borde y un relleno. Los colores del borde y el relleno se pueden definir con los métodos `stroke` y `fill`, respectivamente, cada uno de los cuales recibe como argumento un número entero en [0, 255] codificando el color. Si no se especifica nada, los valores por defecto son 0 (negro) para el borde y 255 (blanco) para el relleno. Además, el color del fondo puede controlarse con el método `background` cuyo valor por defecto es 205.

Prueba ahora a dibujar un círculo negro con el borde gris, centrado en (20, 30) y con radio 10. Para apreciarlo mejor, conviene poner el fondo de color blanco.

Observa que no toda la ventana se ha coloreado en blanco. El motivo es que la ventana contiene un *lienzo* que es donde se realmente se dibuja. El tamaño del lienzo puede controlarse con el método `size` que recibe 2 parámetros: la anchura y la altura. Prueba a crear un lienzo de tamaño 30x30 y repite el ejercicio anterior. ¿Qué sucede ahora?

Ahora trata de obtener algo parecido a lo que se muestra en la Figura 4.

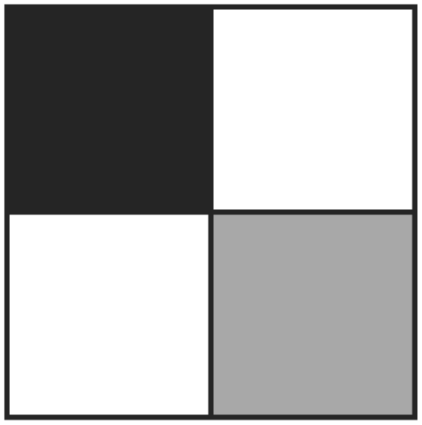

*Fig. 4. Ejemplo de figura en blanco y negro*

El siguiente paso es considerar cualquier color, para lo cual utilizaremos el modelo RGB (del inglés Red-Green-Blue), aunque existen otros modelos para trabajar con colores en informática gráfica. El modelo RGB se basa en la idea de que cualquier color puede representarse a partir de 3 colores primarios: rojo, verde y azul. Es decir, cada color será representando especificando 3 números enteros en el intervalo [0, 255] indicando la cantidad de cada uno de los 3 colores primarios.

Por ejemplo:

- `fill(255, 0, 0)` usa como relleno rojo “puro”
- `fill(0, 255, 0)` usa verde “puro”
- `fill(0, 0, 255)` usa azul “puro”

¿Podrías predecir qué sucederá en los siguientes casos? Piénsalo y compruébalo después.

```
a) fill(255, 255, 0)
b) fill(0, 255, 255)
c) fill(255, 0, 255)
d) fill(255, 255, 127)
e) fill(127, 255, 255)
f) fill(255, 127, 255)
```

Puesto que trabajar con este modelo es muy complicado para los humanos, Processing proporciona la herramienta *"Color selector"*, que permite seleccionar gráficamente un color y obtener sus coordenadas RGB.

Utilizando *"Color selector"*, dibuja un rectángulo morado con el borde pistacho.

Al trabajar con colores se puede añadir un cuarto parámetro, llamado alfa y que corresponde a la transparencia del color, algo útil al trabajar con elementos superpuestos.

Dibuja 2 figuras que se superpongan y comprueba el funcionamiento del parámetro alfa.

# 4. Te presento a Zoog

En la siguiente sesión de prácticas continuaremos profundizando en Processing. El punto de partida será el alienígena *Zoog*, que se muestra en la Figura 5. Intenta escribir las instrucciones adecuadas para dibujarlas y no olvides traer el fichero a la próxima sesión.

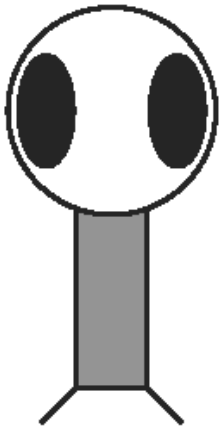

*Fig. 5. Zoog*

# Apéndice 1. Instalación de Processing

Para finalizar esta práctica, se explica el proceso que debe seguir el alumno para instalar en sus ordenadores personales el software Processing, necesario para realizar las prácticas 1 y 2 de Informática. Se deben seguir los siguientes pasos:

1. Acceder a la página web de Processing http://processing.org y seleccionar la sección *"Download Processing"* (ver Figura 6).

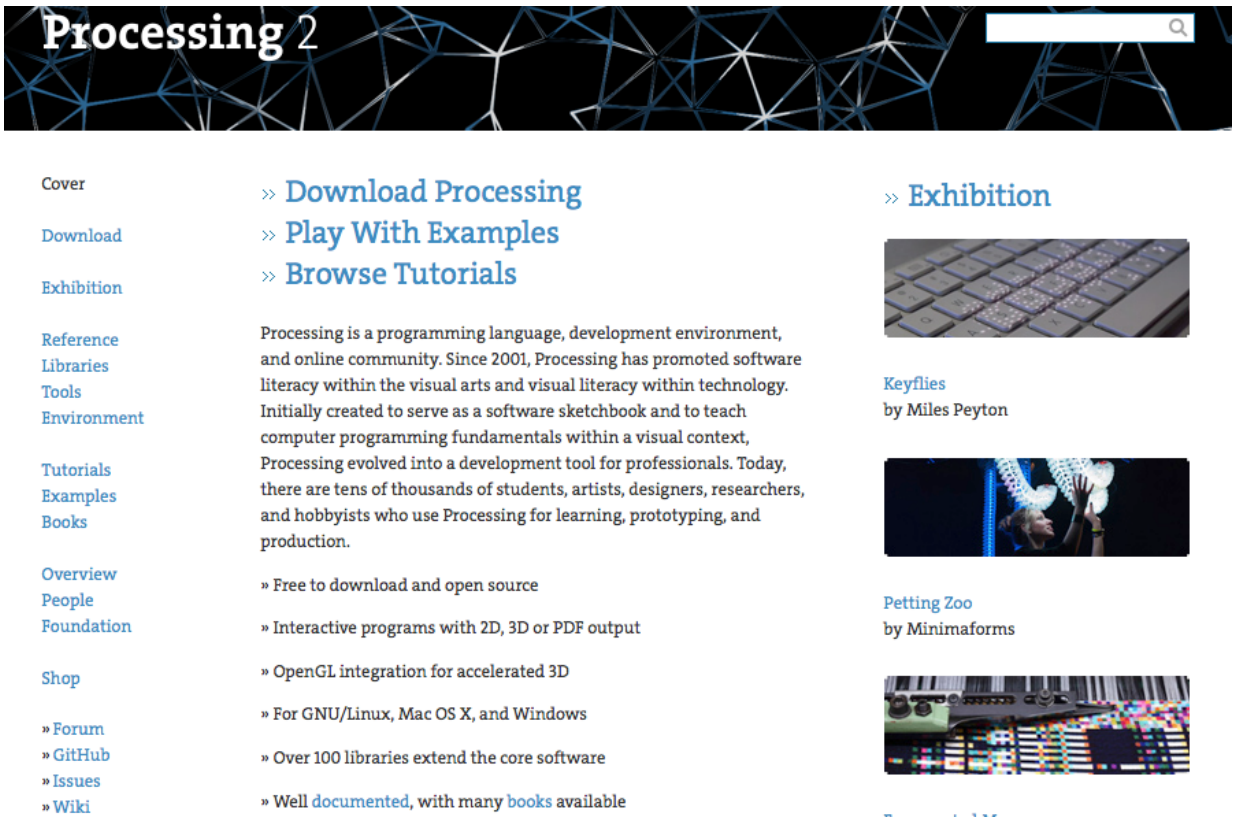


*Fig. 6. Página de Processing*

2. A continuación se ofrece la posibilidad de colaborar con la finalización de Processing a través de una donación voluntaria, si no estamos interesados podemos seleccionar *"No Donation"* y pulsar *"Download"* (ver Figura 7).
3. En ese momento, se deberá pinchar sobre el enlace correspondiente al sistema operativo del ordenador donde se desea instalar Processing (ver Figura 8) y se iniciará la descarga de un fichero con extensión zip.
4. Una vez descomprimido, Processing ya está listo para utilizarse sin necesidad de instalación, bastará con hacer doble click sobre el fichero ejecutable `Processing.exe` (en Windows).

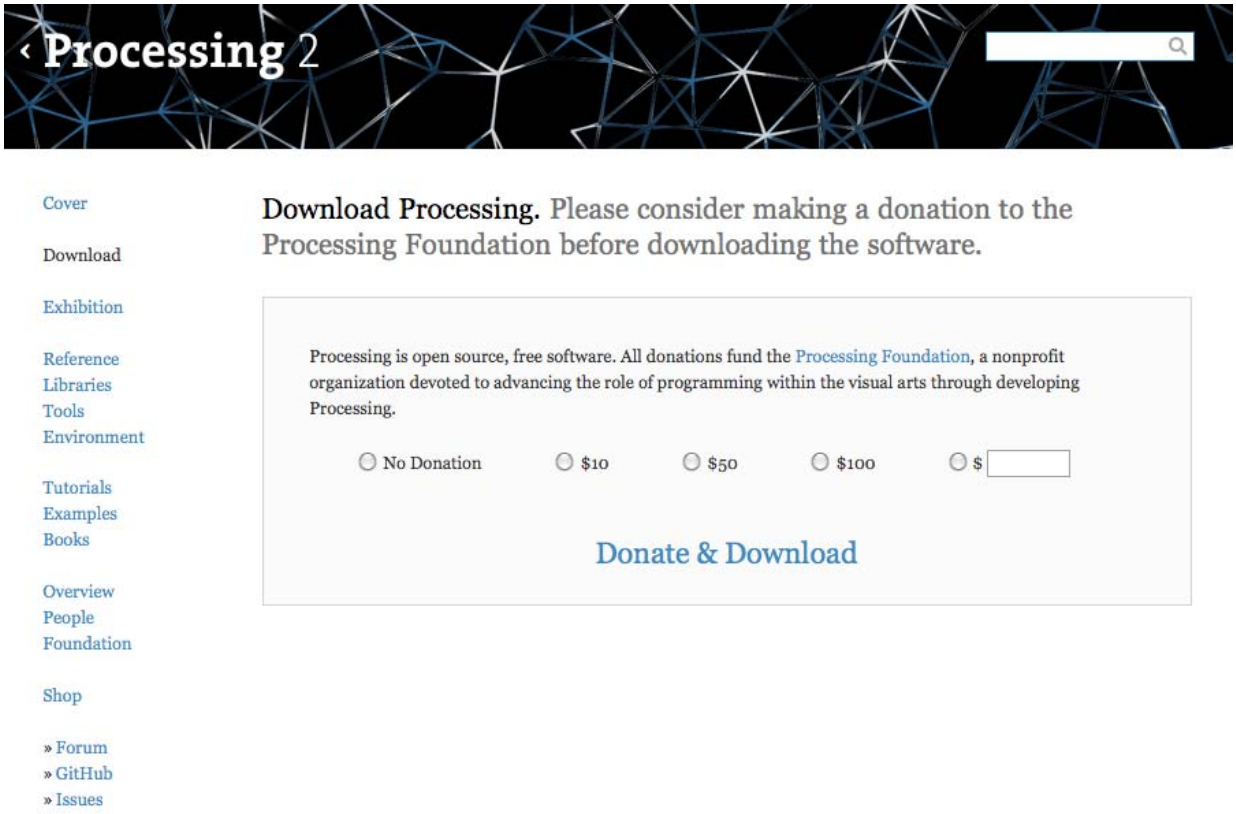


*Fig. 7. Página previa a la descarga de Processing*

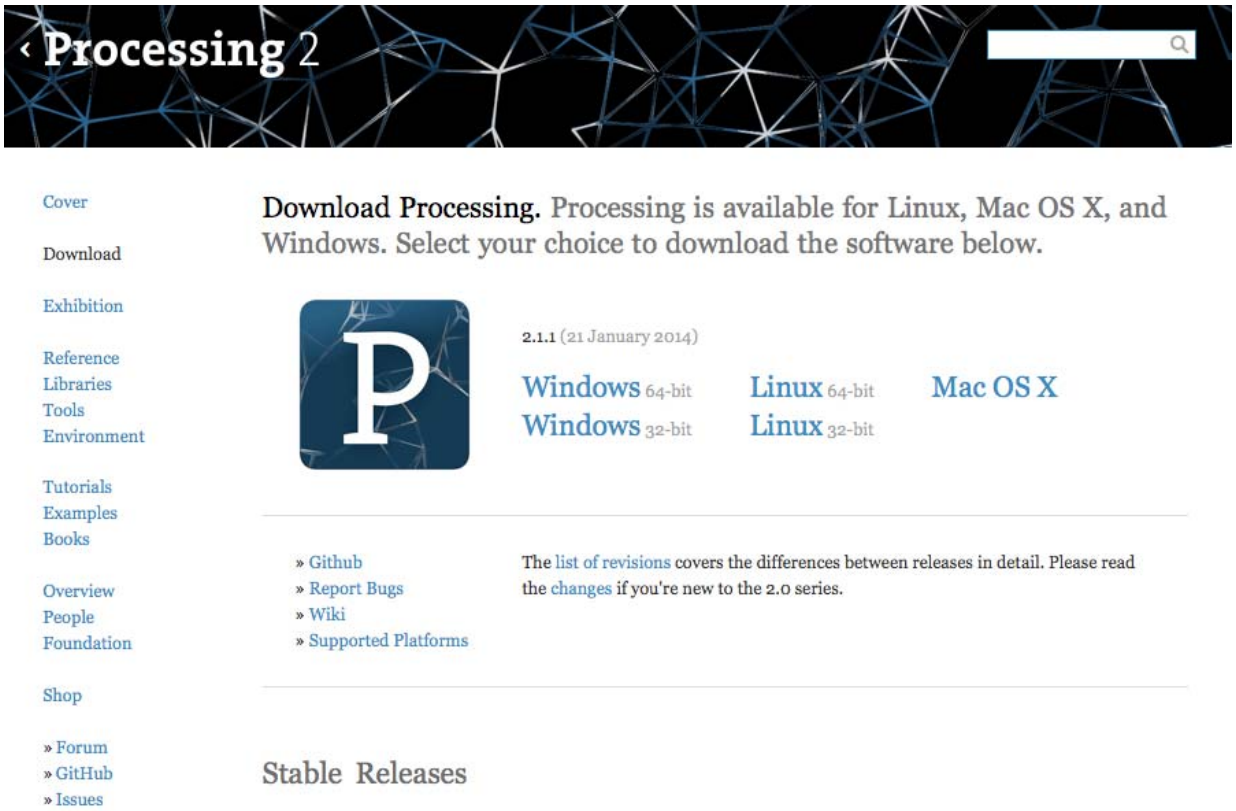


*Fig. 8. Página de descarga de Processing*