

Introducción a Processing

Quelic Berga

PID_00201065



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

1. Introducción.....	5
2. ¿Qué es Processing?.....	6
3. Contexto: del diseño gráfico en papel al diseño computacional.....	7
4. El nacimiento de Processing.org.....	9
5. La comunidad, recursos sobre Processing.....	10
5.1. Bibliotecas de recursos, ejemplos, canales y grupos	10
5.2. Tutoriales	13
5.3. Videotutoriales	14
5.4. Libros	15
5.5. Autores de referencia	17
5.6. Herramientas en línea	17
6. Comunidad de desarrolladores.....	20

1. Introducción

En este capítulo, presentamos el programa Processing. Se presentan sus orígenes, el contexto en el que se desarrolló, recursos y enlaces para ver ejemplos, aprender y ampliar las posibilidades de este.

Figura 1



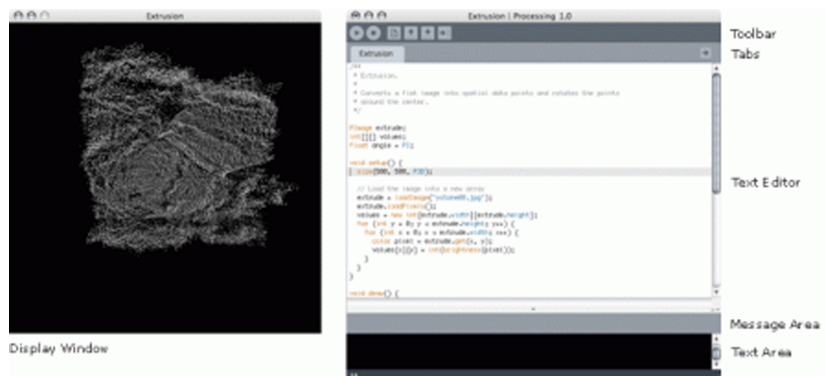
2. ¿Qué es Processing?

Como dice de forma simple y clara la página oficial, se trata de un lenguaje de programación de código abierto para gente que quiere crear imágenes, animaciones e interacciones. Inicialmente desarrollado como software para crear bocetos de trabajo y enseñar los principios de la programación en un contexto visual, Processing ha evolucionado hacia una herramienta de producción profesional. Hoy en día, hay decenas de miles de estudiantes, artistas, diseñadores, investigadores y aficionados que usan Processing para aprender, elaborar prototipos y producir proyectos a diario.

Características del lenguaje:

- es gratuito y de código abierto,
- cuenta con generación de programas interactivos con salida en formatos 2D, 3D y PDF,
- tiene integración con OpenGL para usar aceleración 3D,
- es compatible con GNU/Linux, Max OS X y Windows,
- los proyectos se pueden ejecutar como *applets* de Java en la Red,
- cuenta con más de 100 librerías que extienden el potencial del programa con sonido, vídeo, visión por computador y más,
- está bien documentado y hay varios libros disponibles.

Figura 2



3. Contexto: del diseño gráfico en papel al diseño computacional

Es evidente que no es lo mismo crear un diseño en papel que crearlo para un dispositivo móvil. Asimismo, es evidente que la tecnología afecta a nuestro modo de trabajar, pues hoy podemos hacer ciertas cosas que antes no podíamos, pero también tenemos un contexto mucho más complejo y más acelerado que condiciona la calidad del trabajo.

En la década de 1980, la aparición del ordenador personal con su interfaz gráfica, junto con programas para la generación de carteles, ilustraciones, retoques fotográficos y dibujos empezó a provocar cambios en el sector de las artes gráficas. Surgieron diseñadores gráficos que se adaptaron a las nuevas herramientas, así como retratores que lo consideraron un peligro para la disciplina. También surgieron un grupo creciente de diseñadores que tuvieron su primer contacto con la disciplina mediante los ordenadores o, más específicamente, mediante el uso de las interfaces de los programas.

En todo este creciente maremoto de cambios provocados por las nuevas tecnologías, en la década de 1990, un personaje un tanto peculiar planteó una serie de postulados que influyeron en los actuales creadores de Processing.org. Estamos hablando de John Maeda, actual director de la Rhode Island School of Design, ganador de prestigiosos premios de diseño gráfico y considerado un gurú en el ámbito del diseño y la tecnología.

John Maeda

El japonés-americano John Maeda estudió computación en el prestigioso MIT, pero se doctoró en diseño en la escuela de arte de la Universidad de Tsukuba. Este recorrido cruzado entre ciencias computacionales y diseño gráfico ayudó a Maeda a convertirse en pionero del diseño computacional.

El éxito de herramientas como QuarkXpress, Photoshop o Freehand recayó en que usaron analogías, metáforas e iconos que mimetizaron las herramientas que ya usaban los diseñadores (como las tijeras, la pluma o la lupa). Estas metáforas son muy prácticas para aquellos que ya conocen el oficio del diseñador gráfico y quieren ir más rápido con una nueva herramienta pero, como apuntó Maeda, quizás limita la imaginación y las posibilidades que la computación nos brinda.

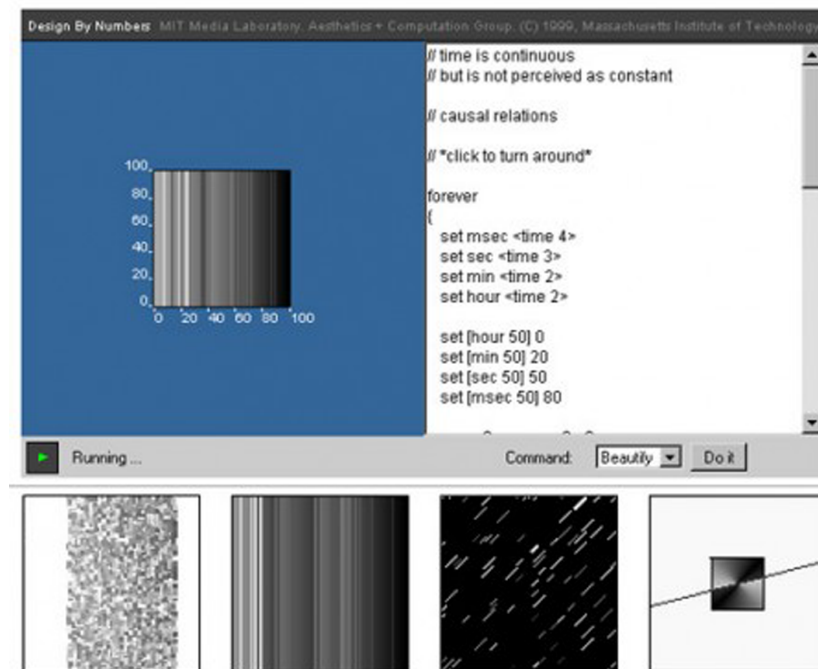
Así pues, entre 1995 y 1999, Maeda publicó *Reactive Books*: cinco libros dedicados a explorar las fronteras del diseño en papel y el diseño en pantalla. Maeda nos habla con melancolía del papel, de la textura, de lo que se ha perdido con las nuevas tecnologías, pero también nos habla de las nuevas posibilidades que se abren al usar la computación en el diseño. En cada libro, elabora una propuesta gráfica y presenta un programa interactivo, así reflexiona a escala teórico-práctica sobre los fundamentos de la interactividad (el tiempo, el teclado, el ratón, el sonido y la webcam/vídeo). En ellos, habla de las distancias entre el papel y la interactividad, pero a su vez juega con la relación que existe entre uno y otra, creando páginas con diseños hechos mediante programación y buscando en la pantalla la calidez y textura del papel.

En 1999, crea un programa para enseñar los fundamentos de la programación/computación a diseñadores gráficos. Se trata de DBN (Design By Numbers), un sencillo programa con una ventana para escribir código de forma simple y usando un léxico parecido al de los diseñadores y otra ventana para ver los resultados. Así, consigue acercar a los diseñadores el mundo de las matemáticas y viceversa y, a su vez, le permite ilustrar a sus alumnos el potencial del diseño computacional.

Reactive Books

Para saber más sobre *Reactive Books*, os recomendamos la visión del vídeo explicativo que podéis encontrar en el siguiente enlace: <http://www.maedastudio.com/2004/rbooks2k/index.php>.

Figura 3



4. El nacimiento de Processing.org

Casey Reas y Benjamin Fry, autores de Processing.org, estudiaron con John Maeda en el MIT y conocieron de primera mano DBN y las teorías de Maeda. De hecho, el propio Reas confiesa en su blog que no habría Processing sin DBN. Para ellos, ver un programa tan simple, compatible con la Red y a su vez estructurado para aprender y compartir el código fue clave. Así pues, respetando el trabajo de Maeda, iniciaron el proyecto de código abierto Processing.org.

En el 2001, dentro del grupo de investigación de Aesthetics and Computation del MIT Media Lab, publicaron la primera versión de este lenguaje junto con un sencillo entorno de programación (IDE). El objetivo inicial de sus autores fue, continuando con el legado de DBN, mejorar el rendimiento y facilitar a artistas y creadores el poder hacer bocetos (*sketches*) de forma ágil y rápida para después desarrollar con otros lenguajes sus proyectos.

La comunidad de artistas encontró en Processing una herramienta para experimentar con visuales interactivos, similar a Flash pero, huyendo de las metáforas, mucho más orientada al código y pensada para dar los primeros pasos en programación. La comunidad de programadores encontró una plataforma fácilmente extensible mediante librerías, bien documentada y activa. Estos dos hechos, junto con la excelente labor de profesores y profesionales que lo usan, han provocado que Processing se esté convirtiendo en una herramienta de referencia para el diseño computacional y el arte generativo.

Estamos a punto de ver el lanzamiento de la versión estable 2.0, que promete mejoras en el rendimiento 3D, compatibilidad y capacidad de exportar a Javascript (y se puede usar en cualquier navegador compatible con HTML5 mediante Canvas), así como capacidad de exportar a aplicaciones Android, además de las ya implementadas exportaciones a Linux, Mac OS X, Windows y *applet* de Java.

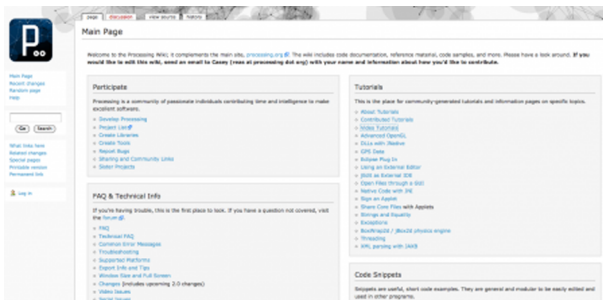
5. La comunidad, recursos sobre Processing

Gracias a la propia naturaleza del proyecto, hablar de recursos de Processing es sin duda apuntar hacia las buenas labores que se están haciendo desde la propia comunidad. Uno de los mejores recursos es la página oficial del proyecto, que además de permitir descargar la IDE, dispone de tutoriales, referencias al lenguaje, ejemplos, comentarios de libros, una selección de proyectos hechos con Processing y enlaces a las comunidades que apoyan el proyecto. A continuación, vamos a comentar algunos de estos recursos, añadiremos algunos en español y proponemos algunos que no están en la página oficial.

5.1. Bibliotecas de recursos, ejemplos, canales y grupos

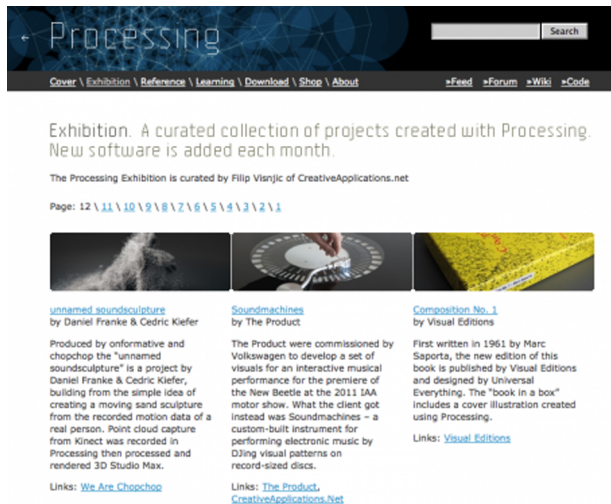
El **primer recurso** por analizar es el wiki oficial. Ahí encontraréis todo tipo de recursos actualizados y, si queréis ir por libre, en este espacio encontraréis todo lo que necesitáis: tutoriales, cómo participar, ayuda técnica, entrevistas, comparativas de lenguajes de programación, entre otros.

Figura 4. Wiki oficial



También dentro de la misma página oficial de Processing podréis encontrar **muestras de proyectos** finales que usan este lenguaje.

Figura 5. Página oficial de Processing



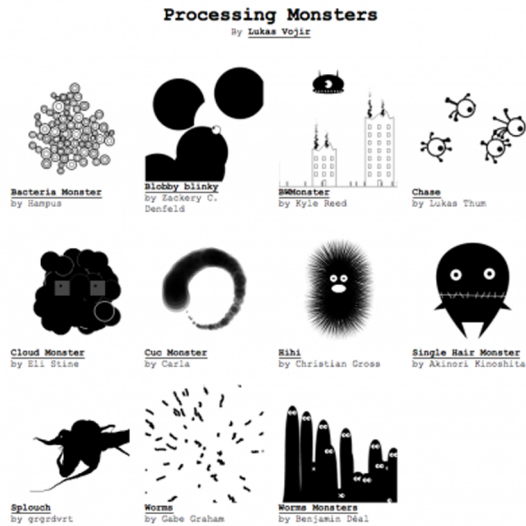
Openprocessing.org es uno de los espacios más interesantes de la comunidad. Los profesores y los centros educativos **pueden generar aulas virtuales** donde compartir los ejercicios con los alumnos. Como se mantiene en abierto, permite ver e incluso ejecutar directamente el código en la web. Los alumnos suben también sus ejercicios y se convierte en un buen sitio para ver muchas propuestas desde distintos puntos de vista.

Figura 6. Openprocessing.org



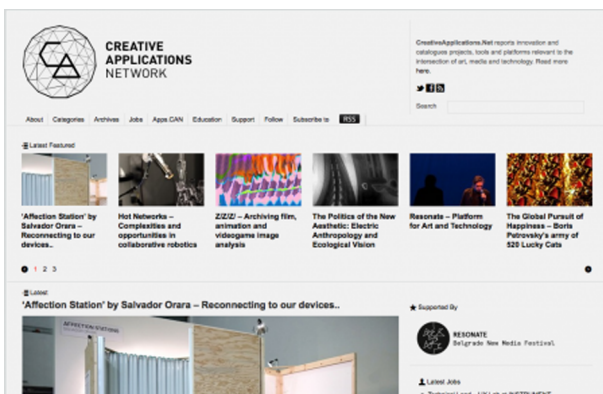
Una **web curiosa e interesante** es la de Lukas Vojir, que nos propone descubrir distintos monstruos en blanco y negro que reaccionan al movimiento del cursor. Como siempre, podéis ver el código, aportar vuestro propio monstruo y navegar en búsqueda de vuestro monstruo favorito.

Figura 7. Web de Lukas Vojir



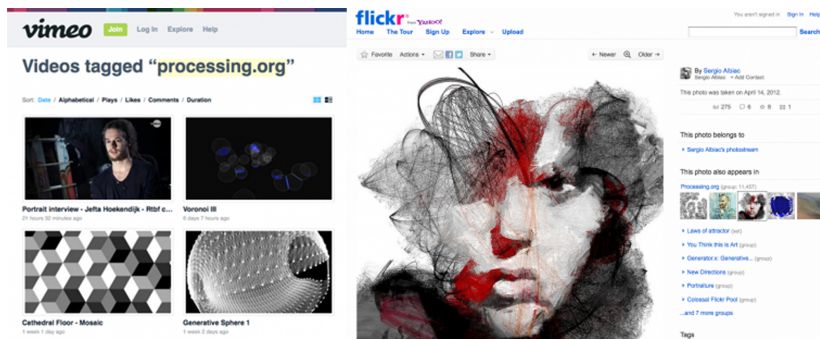
Creative Applications Network es un blog que está haciendo una labor de alta calidad al recoger trabajos interesantes sobre diseño interactivo y creatividad. Os recomendamos echar un vistazo a la categoría dedicada a Processing.

Figura 8. Creative Applications Network



Si lo que queréis es ver ejemplos de lo que hace la gente con **Processing**, lo mejor es que os toméis un tiempo para ver las más de 150 páginas de vídeos en Vimeo con el *tag* [processing.org](https://www.youtube.com/watch?v=processing.org) o que visitéis el grupo de Flickr para ver imágenes generadas con este programa. Recordad que muchas de estas muestras de proyectos vienen con los enlaces a las webs de los autores o con un email de contacto que os puede ayudar a conocer mejor el proyecto e incluso a entrar en colaboración.

Figura 9. Processing en Vimeo y Flickr



5.2. Tutoriales

De nuevo, un buen sitio para empezar con tutoriales es la misma **página de Processing**, con el apartado dedicado al aprendizaje. Al usar el IDE de Processing, si seleccionáis una palabra, con el botón derecho veréis la opción de ir directamente a la referencia del lenguaje junto con algunos ejemplos.

Otro recurso muy interesante para empezar es **la web de Daniel Shiffman**, que a su vez tiene un libro. En ese espacio, encontraréis ejemplos, tutoriales y ejercicios para hacer un pequeño curso en línea y si preferís basaros en una versión traducida al español podéis encontrar un tutorial en castellano.

Figura 10. Learning Processing

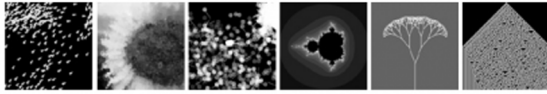


Si no lleváis muy bien las lenguas, podéis encontrar **un interesante tutorial** realizado por Joan Soler en castellano.

Si lo que realmente os impresiona son la física y las matemáticas, entonces estáis de suerte porque Daniel Shiffman está desarrollando un nuevo proyecto basado en este tema para **aprender Processing**, así que podéis conseguir dos cosas al precio de una... y siempre en abierto y gratuito.

Figura 11. Nuevo proyecto de Daniel Shiffman

The Nature of Code



I'm please to announce that I am working on expanding all of these tutorials into a book. Support the project (and get advance PDFs) on [Kickstarter!](#)

Spring 2011 Syllabus: <http://itp.nyu.edu/varwiki/Syllabus/Nature-of-Code-S11>

5.3. Videotutoriales

Si lo vuestro no es leer sino mirar, y si andáis un poco perdidos con los conceptos básicos de programación, esta completa web os ayudará a empezar de cero y lograr buenos resultados con ejemplos, vídeos y ejercicios.

Figura 12. Fun programming

Fun Programming

Videos for learning creative programming

1. Introduction



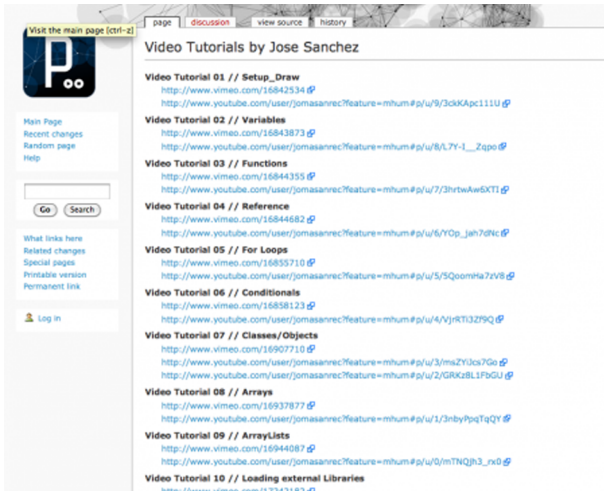
Welcome to Fun Programming! This is the introduction to a series of videos where I will be explaining creative programming to beginners. I start with a programming language called Processing in the next video. These videos are part of a challenge to upload one episode each day during August 2011. I focus on creating visual effects and not so much in theory. Usually I introduce one new concept on each episode. Most images in this video are taken from the programs we write in coming episodes. In www.funprogramming.org you can read the descriptions of all episodes in one page. Note: This video replaces the original introduction video. It's the same but better :)

2. Download Processing. Use `point()` and `line()`

Download Processing from [www.processing.org and de.com](#)

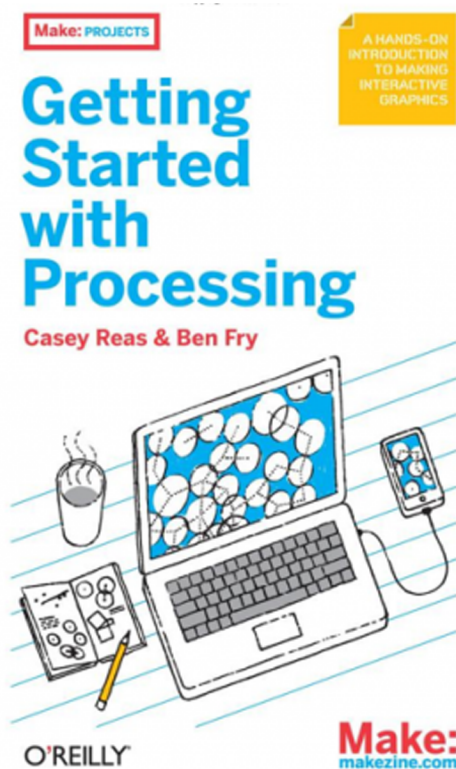
José Sánchez tiene una serie de videotutoriales colgados en Vimeo y en YouTube; para encontrar un listado completo de esos vídeos, podéis ir directamente al wiki de la página de Processing.

Figura 13. Videotutoriales



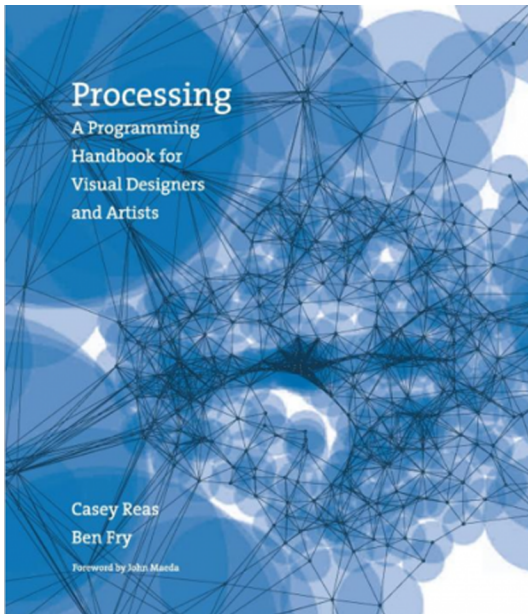
5.4. Libros

Podéis encontrar referencias de prácticamente **todos los libros de Processing** en la sección *books* de su página web. Según vuestros intereses y necesidades os será más útil uno u otro. Si estáis empezando con el tema de programar, os aconsejamos el último que ha publicado O'Reilly Media, de los propios autores de Processing y con un tono claramente informal y próximo.

Figura 14. Casey Reas; Ben Fry (2010). *Getting started with processing*. O'Reilly Media.

De los mismos autores, tenemos **una guía más extensa** con un prólogo de John Maeda y con una interesante introducción que habla de cómo la computación y el arte llevan desde el principio de la mano. El libro es el resumen de la experiencia pedagógica de los autores con tutoriales, ejemplos y pequeñas explicaciones teóricas que os darán un conocimiento sólido.

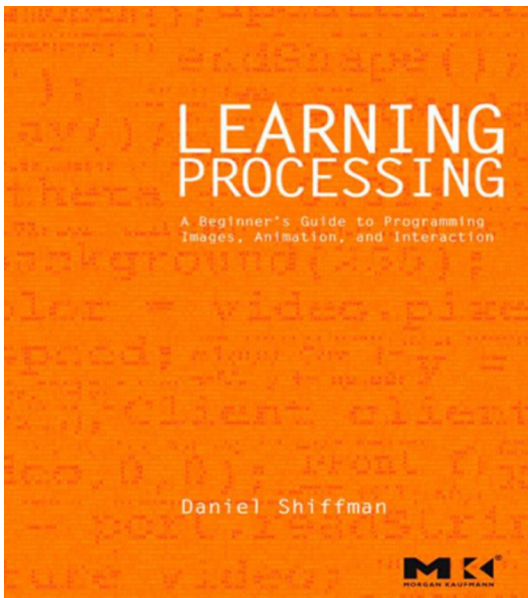
Figura 15. Casey Reas; Ben Fry (2007). *Processing: a programming handbook for visual designers and artists*. MIT Press.



Ira Greenberg llevó a cabo una excelente labor con este libro, parecido al comentado antes, y en este caso publicado por la editorial Friends of ED, especializada en diseño, física y programación. Este libro está enfocado a **perfiles pluridisciplinarios** que sienten interés por el diseño, el arte, las matemáticas, la programación y parte siempre de cero para, en poco tiempo, empezar a hacer ejercicios prácticos. *Processing: creative coding and computational art*, de Ira Greenberg, publicado por Friends of Ed.

Antes hemos hablado de un tutorial de **Daniel Shifmann para aprender Processing**, aquí tenéis los datos del libro que acompaña y extiende el proyecto web.

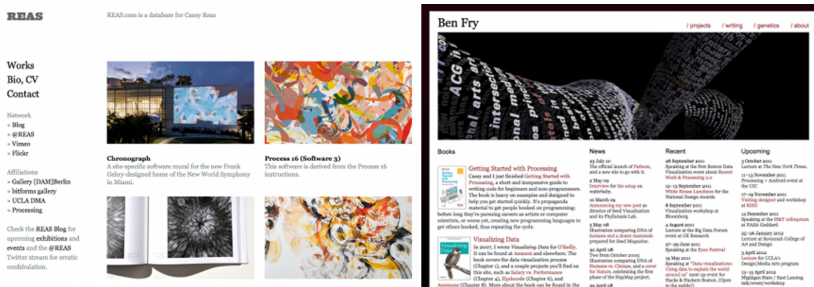
Figura 16. Daniel Shiffman (2008). *Learning Processing: a beginner's guide to programming images, animation, and interaction*. Morgan Kaufmann.



5.5. Autores de referencia

Hemos hablado de los creadores de Processing. Seguramente, si queréis seguir el proyecto de cerca y ver su faceta más artística, lo mejor será echar un vistazo a las páginas personales de Casey Reas y Ben Fry.

Figura 17. Páginas de Casey Reas y Ben Fry



También es interesante, sobre todo porque es uno de los desarrolladores de librerías y escritor de documentos más activo de la comunidad, la web de Daniel Shiffman.

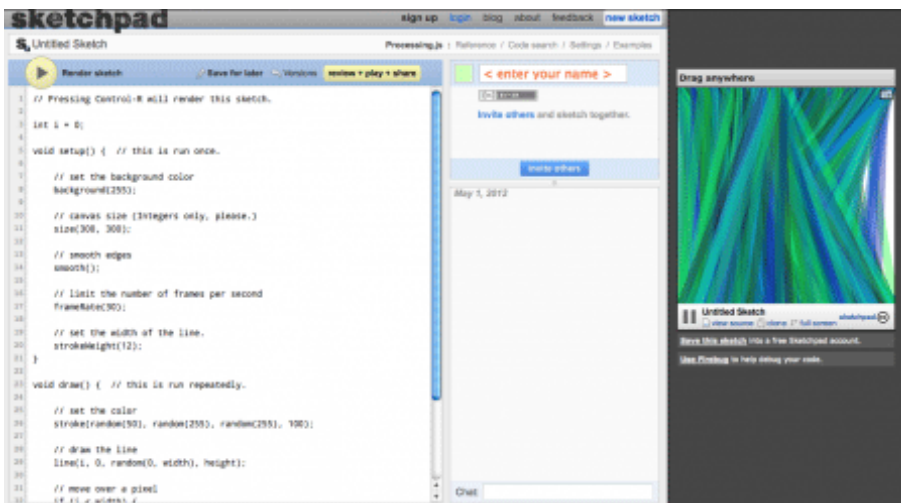
5.6. Herramientas en línea

El siguiente enlace permite ejecutar código directamente en la web y ver los resultados en directo: <http://sketch.processing.org/>.

Figura 18. Ejemplo de ejecución de código en *sketch processing*

```
1 // comments go here
2
3 void setup()
4 {
5   size(200,200);
6   background(125);
7   fill(255);
8   noLoop();
9   PFont fontA = loadFont("courier");
10  textFont(fontA, 14);
11 }
12
13 void draw(){
14   text("Hello Web!",20,20);
15   println("Hello ErrorLog!");
16 }
17
```

La página de Sketchpad permite, además de escribir en tiempo real, hacerlo de modo colaborativo en un *pad* con la opción de chatear con los contribuidores en el mismo momento que se va modificando el código. Es realmente interesante si queréis colaborar en la distancia con un equipo de programadores.

Figura 19. Ejemplo de ejecución de código en *sketchpad*

Ya lo hemos comentado antes, pero Openprocessing no es solo una comunidad de aprendizaje, sino también una forma de testear los *sketches* en tiempo real.

Figura 20. Openprocessing



En este caso, Processing.js no es una herramienta en línea, sino una librería de Javascript desarrollada por John Resig para que desde Canvas se pueda ejecutar Processing; se logra incrustar en cualquier web vuestros bocetos sin usar ningún *plug-in* adicional.

Figura 21. Processing.js



6. Comunidad de desarrolladores

Y, finalmente, si todo esto os ha sabido a poco, entonces vuestro recurso favorito será entrar por la puerta grande y conocer y participar con la comunidad de desarrolladores.

Figura 22. Comunidad de desarrolladores de Processing

