

Patín: aprendiendo programación de la manera divertida

Autor: Hugo Ruscitti
Docente: Juan Carlos Rosman
Revisiones: Javier de Innocenti

Concepto: Anexo de Trabajo Práctico
Curso: Sábados, 2011 2C
Materia: Creatividad

Contenido

Resumen.....	3
Motivación: ¿Por qué patín?.....	3
Sobre programar y computadoras	3
¿Qué es un programa?	4
De usuarios a desarrolladores.....	4
El proceso de desarrollo.....	5
Creando Patín	5
Diseñar haciendo preguntas y tomando riesgos	5
Diseño de la interfaz:.....	6
Al ejecutar.....	6
Panel lateral.....	7
Diseñando documentación	8
Más información.....	9

Resumen

Patín es una herramienta para que los chicos puedan aprender a programar computadoras de manera práctica, divertida, e interactiva.

Haciendo programas se puede aprender sobre creatividad, computadoras, proyectos y ciencia. Muchos investigadores están de acuerdo en que construir proyectos dentro del aula representa una oportunidad muy valiosa para aprender y crecer.



En este documento intentaré mostrar algunos conceptos de diseño que usé para crear la aplicación Patín, el contexto educativo en donde podría funcionar y otras ideas interesantes para el futuro del proyecto.

Motivación: ¿Por qué patín?

Cuando nuestro profesor de creatividad, Juan Carlos Rosman, nos dio la posibilidad de elegir la temática de trabajo práctico comencé a crear un prototipo de lo que ahora es un programa que funciona (mas o menos bien), y que tiene un potencial muy alentador.

En este trabajo anexo intentaré ordenar un poco mis ideas, detectar los conceptos clave de la motivación para escribir esta herramienta y contar parte del desarrollo de algunos componentes.

Sobre programar y computadoras

Muchas personas ven a la programación como algo frío, ordenado y exacto, observan a las computadoras como algo que viene del mundo de la ciencia teórica, que son cosas difíciles de entender y solo las pueden controlar los adultos científicos. En ese paradigma, muchos chicos pierden el interés por la programación sin siquiera comenzar¹.

Pero en realidad, se trata de un mapa equivocado, hacer programas de computadora es muy divertido y te da muchas herramientas y oportunidades de aprendizaje: te permite tener el control de la computadora y decirle qué hacer. Podrías crear un videojuego, una aventura multimedia, un sitio web o un puñado de ventanas con botones e imágenes que hacen algo útil (o no).

Es aquí donde nace mi motivación con las computadoras, me gustaría que mas personas tengan la posibilidad de ver a la computadora como una herramienta útil, para construir y divertirse *creativamente*.

No es cierto que las computadoras sean difíciles de programar o que se requieran conocimientos de matemáticas, la computación es un concepto mucho mas amplio, y curiosamente, tiene mas similitudes con el arte² de las que imaginamos a primera vista.

1 Roberto Alsina escribió sobre esta imagen que se genera cuando no entendemos sobre una determinada actividad: programar, matemáticas, o incluso tejer. Su artículo es muy interesante y recomendable: <http://lateral.netmanagers.com.ar/tr/es/weblog/posts/BB954.html>

2 Comúnmente se asocia a la programación y los sistemas con las ciencias, pero una parte importante del desarrollo de sistemas sigue prácticas mas similares al mundo de la pintura que al de las ciencias. El texto *Hackers y Pintores* es una excelente lectura complementaria para analizar este punto: <http://hackerdou.com/?p=777>

¿Qué es un programa?

Un programa de computadora es una serie de instrucciones que sirven para decirle a una máquina qué tiene que hacer.

Un programa se ve muy parecido a una receta, tiene ingredientes y un paso a paso para llegar a elaborar un resultado.

Básicamente, cuando la computadora tiene un programa para ejecutar, va a ir paso a paso interpretando nuestro código y siguiendo todas las órdenes que le demos.

Lo interesante de las computadoras, a diferencia de un electrodoméstico, es que la computadora puede leer programas que nosotros escribimos, y no solo los programas que se instalaron de fábrica.

De usuarios a desarrolladores

Hacer programas de computadora no es solamente escribir código, consiste en crear una estrategia paso a paso de lo que queremos lograr. Es algo dirigido a un resultado, ir desde una idea básica (a veces incompleta) hasta convertirla en realidad.

Por ese motivo, es una habilidad muy potenciadora, les da el poder a los chicos para que controlen la computadora y tengan un papel mas activo con la tecnología.

Las computadoras y los programas (cómo los videojuegos) están en la cultura de los chicos, es corriente verlos hablar de juegos, de aplicaciones o estrategias para jugar. Por ese motivo, es importante que tengan la oportunidad de aprender de qué están hechos los programas, que se puedan acercar para comprender que detrás de las máquinas hay ideas, proyectos y decisiones (a veces incorrectas³) de cómo tienen que funcionar los programas.

El objetivo mas ambicioso de *patín*, es ayudar a que los chicos se animen a expresar su creatividad. Que los ayude a pasar de usuarios a desarrolladores, que puedan ver a la computadora como algo menos misterioso, y que vean a los programas cómo algo que se construye con ideas y un poquito de conocimiento práctico, algo que se puede hacer si uno se lo propone.

³ Joel, uno de los creadores de *stackoverflow*, escribió sobre cómo se toman ciertas decisiones en aplicaciones muy populares:

<http://spanish.joelonsoftware.com/uibook/chapters/3.html>

El proceso de desarrollo

Creando Patín

Existen muchas formas de escribir programas, pero hay ciertos obstáculos que requieren un rediseño de las herramientas tradicionales para acercarse a los chicos.

Patín es un programa que escribí buscando encontrar una manera distinta de que los chicos se puedan acercar a la programación de manera simple y significativa.

Usé dos herramientas llamadas *python* (un lenguaje) y *Qt* (un kit de componentes visuales) junto a varias ideas inspiradas en el constructivismo de [Seymour Papert](#) y otros proyectos que iré citando a lo largo del texto.

Diseñar haciendo preguntas y tomando riesgos

La mayoría de los programas gráficos suelen usar metáforas comunes, tradicionales. Es habitual ver programas que hablan de archivos, menús, ventanas e iconos.

Pero para escribir *Patín*, intenté abordar un ejercicio básico que aprendí del programador *mystilleef*⁴.

Este ejercicio consiste en hacerse preguntas: ¿Cuál es el propósito de la interfaz?, ¿Por qué los programas necesitan que les digamos cuando grabar?, ¿Por qué las aplicaciones tienen menús?.

Encontré este ejercicio clave para hacer Patín, porque haciéndote estas preguntas observas las cosas de otro modo, y te da la posibilidad de tomar riesgos y hacer cosas mas acordes a tus necesidades.

Acá hay algunas conclusiones a las que llegué antes de comenzar a escribir el programa:

- La interfaz tiene que ser facilitadora.
- La creatividad es innata a los chicos, no se debe obstaculizar.
- La experiencia de escribir tiene que ser simple y funcional al mismo tiempo.
- Configurar aplicaciones es aburrido, muchas opciones no es una buena idea.
- No debería ser necesario explicar cómo usar el programa, incluso si sos principiante.
- La ayuda tiene que ser una guía para programar, no un diccionario de cosas desconectadas.

⁴ *mystilleef* es un magnifico desarrollador de software, no solo por la calidad de su código, sino porque escapa a los lugares comunes en el diseño de software. Te recomiendo investigar acerca de su obra *Scribes* en la siguiente web:

<http://scribes.sourceforge.net/index.html>

Diseño de la interfaz:

Teniendo en cuenta las conclusiones mencionadas anteriormente, intenté crear una interfaz simple, pero que tenga la funcionalidad mínima y necesaria para comenzar:

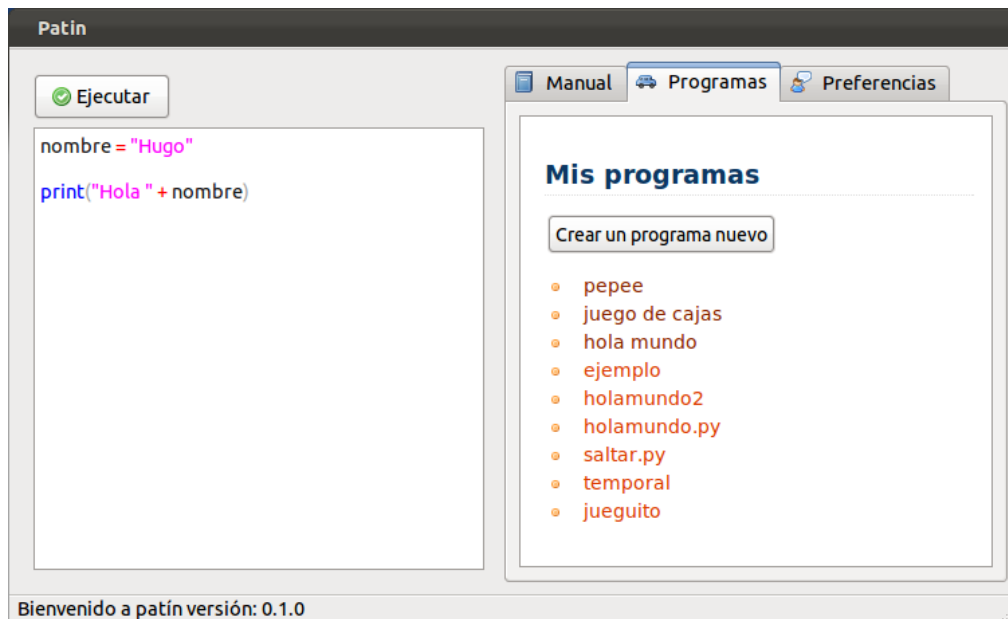


Illustration 1: Patín mostrando la lista de programas y el editor.

La parte izquierda de *Patín*, es un editor que permite escribir y modificar código fuente. En este editor, las palabras clave y las estructuras de programación aparecen destacadas con colores, lo que se conoce con el nombre de *syntax highlight*.

Justo arriba del editor, hay un botón para ejecutar el código ingresado mas abajo y ponerlo a funcionar.

No hace falta preocuparse por el estado del documento, ni de guardar el programa. El botón ejecutar hace esas cosas por su cuenta y elimina inconsistencias de uso.

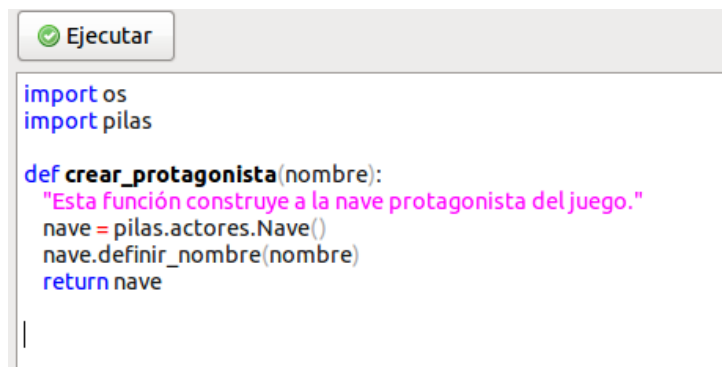


Illustration 2: Syntax Highlight

Al ejecutar...

Cuando se pulsa el botón "ejecutar" ocurren dos cosas, se evalúa el código completo y luego se pasa a un modo interactivo de ejecución mostrando el resultado del programa.

En este modo, se ejecutará cada sentencia que se escriba. Y todo eso dentro del contexto del programa, manteniendo con vida a los objetos y las variables.

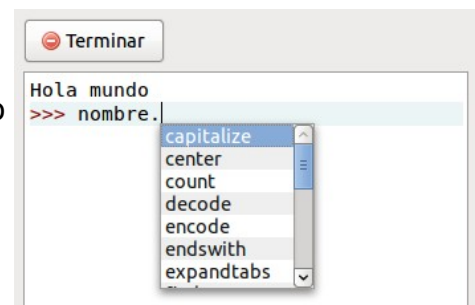


Illustration 3: Entorno interactivo con autocompletado

Esta forma de trabajar es bastante práctica y divertida, porque te permite pasar rápidamente a la acción, realizar pruebas, rediseñar y evaluar conceptos.

Es importante observar, que muchas veces un entorno de prueba es la oportunidad clave para rediseñar o cambiar el curso de un programa. Es una instancia de reflexión y te permite descubrir ideas nuevas e investigar.

Para facilitar la escritura del código, en el modo interactivo se muestra un menú que *autocompleta* palabras inspeccionando el código. De esta manera se fomenta la experimentación y la idea de animarse a descubrir por cuenta propia todas las posibilidades que ofrece cada objeto. Además de reducir al máximo los errores de escritura (y la frustración que eso suele producir).

El modo interactivo también se comporta un poco diferente cuando se trata de un juego, porque cuando tiene que mostrar el resultado del programa, la ventana se divide en dos, y en la parte de arriba se muestra el juego y debajo el entorno interactivo.

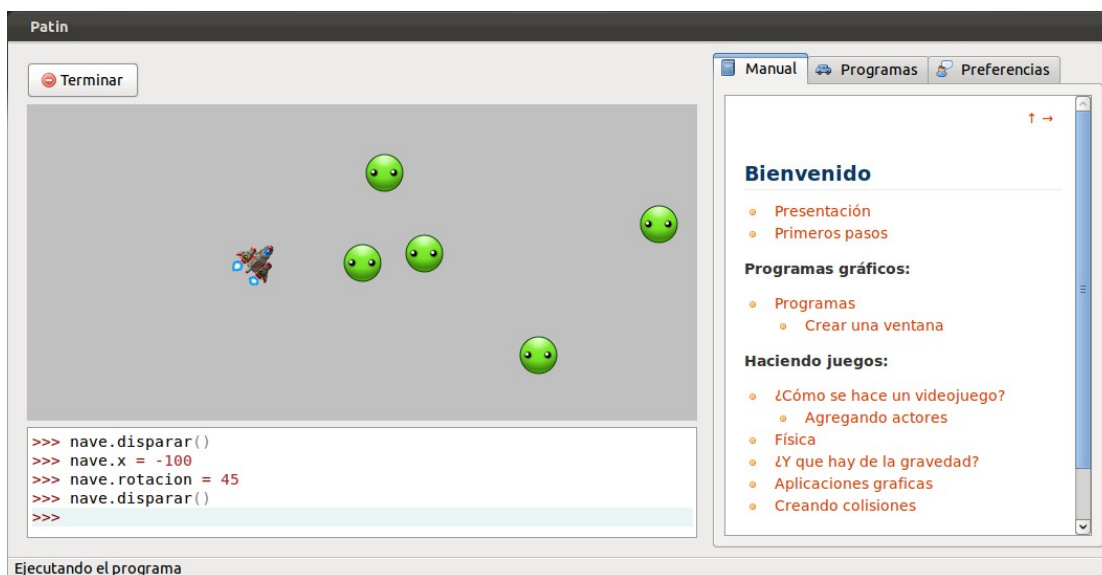


Illustration 4: Patin en modo interactivo creando un juego

En este modo, también se puede usar el teclado y el *mouse* para interactuar con el escenario del juego. Así se combinan dos cosas que tienen mucho en común: jugar y desarrollar.

Panel lateral

En la parte derecha, hay un panel con 3 pestañas: una guía de programación, una lista de programas y un panel de preferencias personales.

La siguiente pantalla muestra el editor junto a la pestaña de opciones:

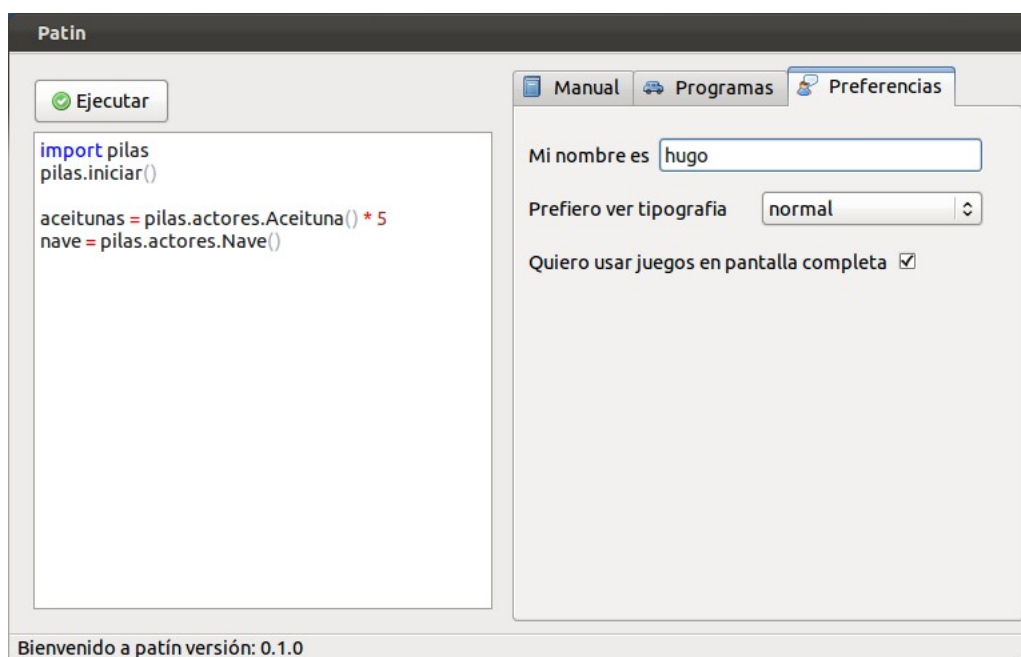


Illustration 5: Pestaña de propiedades

Como se puede ver, la cantidad de opciones y espacio requerido para configuración está muy acotado. La idea de la interfaz es facilitar un modo de trabajo simple y confortable desde el primer día.

Diseñando documentación

Patín ofrece un pequeño manual de programación muy práctico a la derecha del editor. Esta funcionalidad está dirigida a que un chico pueda comenzar a explorar y aprender programación haciendo programas.

El manual es una guía, no un glosario, así que cada ejemplo y cita a elementos del lenguaje se pueden ejecutar, modificar y adaptar fácilmente.

La filosofía detrás del manual, es estructurarlo de modo que sea posible aprender partes simples de programación (cómo funciones) y usarlas tan pronto como se pueda para hacer algo útil. Esta es una de las ideas poderosas⁵ que cita *Papert*.

A medida que se lee la documentación, se pueden ir incorporando mas ideas y conceptos para alcanzar objetivos mas avanzados.

El manual está construido usando un modelo similar a un wiki, usando una herramienta llamada [sphinx](http://sphinx.org), pero integrada a la aplicación como si se tratara de un mini-navegador web.

⁵ Papert, a quien cité anteriormente, es uno de los referentes mas importantes para mí. Uno de los conceptos de su libro “La familia conectada”, ideas poderosas, se puede ver en la siguiente web: <http://fundacionqt.org/blog/?p=830>

Lamentablemente, sus libros son difíciles de obtener, ya no se consiguen en librerías. Así que (descaradamente) te digo: ¡ya sabes lo que puedes regalarme!

Más información

Si quieres conocer mas sobre el curso del proyecto te invito a ver el sitio web de *patín* en www.patin.ep.io y a enviarme *email*: hugoruscitti@gmail.com

Patín arrancó con este trabajo práctico, y creo que tiene potencial para convertirse en un proyecto de [Software Libre](#) colaborativo muy valioso para los chicos, así que me gustaría mucho que te animes a compartir ideas, modificar el programa, investigarlo y a ponerte en contacto.

Para finalizar, quiero agradecerte por leer este documento, me tomó tiempo escribirlo pero valió la pena, ojalá lo hayas disfrutado.

¡Gracias!