

# Pixels & Code

SEPTIEMBRE 2011 > N° 06

LOS SECRETOS  
DE JQUERY

DESARROLLANDO  
APPS PARA BLACKBERRY

The screenshot shows a game interface for 'Bakery'. At the top, there are navigation buttons: 'Free Gifts', 'Play!', 'Invite Friends', 'Get Coins & Cash', and 'Earn More Cash'. The player's score is 60,804, and there are 11 coins. A progress bar is visible. On the left, there are icons for a smartphone (7 notifications), a left arrow, and a right arrow. The main scene features a bakery building with a sign that says 'BAKERY'. At the bottom, there is a control panel with a 'RUSH!' button, a timer (00:00:19), and 'Instant Delivery?' status. The control panel includes sliders for 'View Delivery', 'Beauty' (1/2), 'Size', and 'Truck'. There are also icons for a truck, a person, a trophy, a bar chart, a card, and a 'FRIENDS' button.

## APRENDIENDO A PROGRAMAR VIDEOJUEGOS

Te enseñamos a crear tu primer juego con HTML5/Javascript y Game Maker

# Plataforma para Resellers

## ¡El negocio lo hacés vos!

Tu propio negocio de hosting con todo lo que necesitás.  
Con más de 8 años de experiencia, entendemos tus necesidades. Por eso creamos una plataforma con productos y herramientas para que tu negocio crezca con la mayor rentabilidad.

[www.dattatec.com/resellers](http://www.dattatec.com/resellers)



# Con .TV la gente te ve.

Si tenés **videos**,  
la mejor forma de mostrarlos  
es con un **dominio .tv**

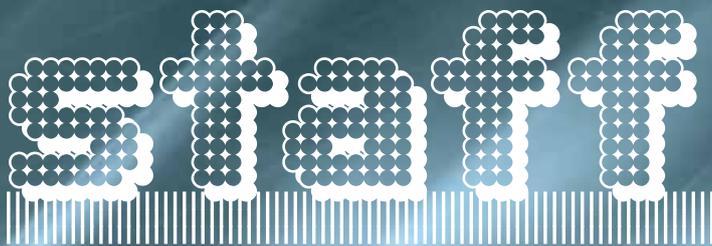
Entrá a **[www.lagente.tv](http://www.lagente.tv)** y registrá el tuyo.

la  
gente **.tv**



[dattatec.com](http://dattatec.com)

Los especialistas en dominios



dattatec.com  
Tu Hosting hecho Simple!

## **JEFA DE REDACCIÓN**



*Débora Orué*

## **COLUMNISTAS**



*Adrián Ortiz  
Cristian Hernán Gaitano Ornia  
Gabriel Walter Gaitano Ornia  
Juan Gutmann  
Roxana Miguel  
Natalia Cantero*



---

## **DISEÑO EDITORIAL Y CREATIVIDAD**

*[www.ampersandgroup.com.ar](http://www.ampersandgroup.com.ar)*

## **REDACCIÓN**

*[lectores@pixelscode.com](mailto:lectores@pixelscode.com)*

## **COMERCIALIZACIÓN**

*Dattatec.com SRL - Córdoba 3753, Rosario, Santa Fe  
[www.dattatec.com](http://www.dattatec.com)*

## **DATOS DE CONTACTO**

*Dirección Comercial:  
[publicidad@pixelscode.com](mailto:publicidad@pixelscode.com)*

*Las opiniones expresadas en los artículos son exclusiva responsabilidad de sus autores y no coinciden necesariamente con la opinión de Dattatec.com SRL.*



**6 // PROGRAMACIÓN VIDEOJUEGOS (I)**  
*Programación de videojuegos*

**14 // BASES DE DATOS (I)**  
*Introducción a las bases de datos relacionales (VI)*

**22 // INVESTIGACIÓN**  
*e-Voto, el diseño del nuevo universo electoral*

**28 // JAVASCRIPT**  
*Conociendo la librería javascript estrella: Los secretos de JQuery*

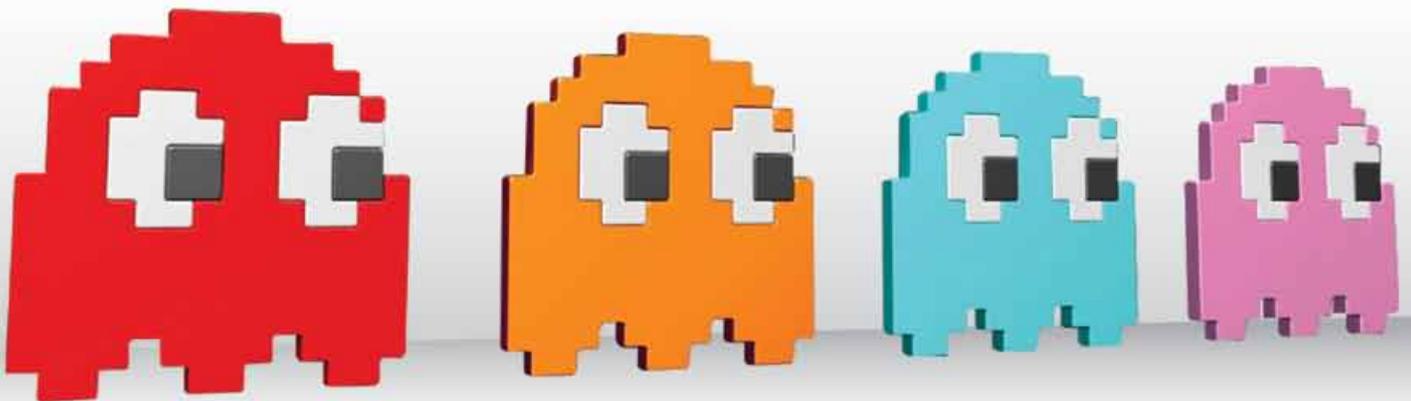
**40 // ALGORITMOS**  
*Algoritmos: Binary Tree Sort*

**47 // PROGRAMACIÓN PARA MÓVILES**  
*Guía para desarrollar aplicaciones para BlackBerry*

**52 // BASES DE DATOS (II)**  
*Atributos decorativos y Reflection*



# Programación de videojuegos



Por Fernando Luna

fernando.luna@pixelscode.com

*El fascinante mundo de la programación engloba diversas áreas: desde los sistemas operativos, drivers, controladores de dispositivos externos, sistemas administrativos, automatización de procesos y hasta videojuegos, cada uno de estos mercados tiene su propio secreto. En esta oportunidad elegimos el apasionante mundo de los videojuegos para tratar de cerca algunos de los secretos más importantes para convertirse en un programador de aventuras gráficas.*

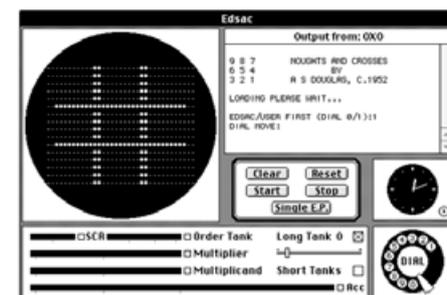
**N**o cabe duda alguna que todos los que utilizamos una computadora para trabajar, estudiar o divertirnos, en algún momento sentimos curiosidad por jugar alguno de los diversos juegos que existen en el mercado. Juegos para Windows, juegos a través de Internet, juegos para videoconsolas o juegos para celulares, por alguno de ellos seguramente pasamos un

buen rato dispersando nuestro ocio y descargando energías acumuladas por obligaciones laborales o académicas. Y seguramente también quienes nos desempeñamos en el mundo de la programación, en alguna oportunidad sentimos la curiosidad de saber cómo llevar adelante una aventura gráfica. Esta nota apunta a enseñar los aspectos básicos de cómo volverse un programador de videojuegos y no morir en el intento.

## UN POCO DE HISTORIA

El pasatiempo de los videojuegos se remonta a muchos siglos atrás. Las partidas de ajedrez, las damas, el famoso juego oriental GO, y otros tantos, inspiraron a muchos hombres abocados a la electrónica de avanzada, en el siglo XX a desarrollar versiones electrónicas de juegos conocidos y deseados por muchos. Si bien quien se interesa por lo técnico, tiene en claro que el primer videojuego lanzado al mercado fue el PONG, esto no es tan así.

1947 es el año en donde se registró la primera patente relacionada a videojuegos. La misma habla de un juego de simulación de misiles utilizando un tubo de rayos catódicos. Unos años más tarde, en 1952, Alexander S. Douglas, quien trabajaba para la Universidad de



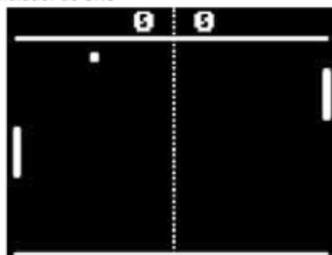
Emulador de OXO



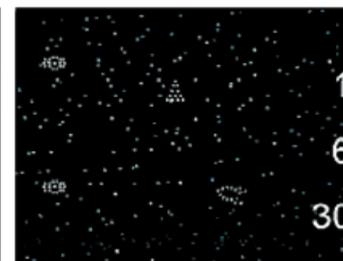
Tennis For Two



Space War



Pong



Asteroids

Cambridge, programó un juego algo adictivo como parte de una investigación que se llevaba a cabo para el estudio de la interacción de las máquinas con los seres humanos. El juego programado fue la versión electrónica del conocido 3 en línea, bautizado por este investigador como OXO. El juego corría solamente en las computadoras ADSAC, las cuales se utilizaban solamente en universidades.

En 1958 William Higinbotham creó el conocido videojuego Tennis for Two, simplemente utilizando una línea horizontal y una pequeña línea vertical, que hacía a su vez de malla. En la pantalla se desplazaba una especie de pelota, comandada por lo que se conocía como Player1, rebotando en otro extremo. La computadora por su parte era la

*Algunos de los clásicos mencionados, culpables con gusto de muchas de nuestras horas infantiles dedicadas frente a un televisor o en las casas de videojuegos por fichas.*

encargada de deducir el recorrido de la bola, que Player1 había generado para devolver el rebote a este jugador.

Unos años después, en 1962, Steve Russel creó un videojuego al cual bautizó Space War. Luego, en 1972, Nolan Bushnel fundó la empresa Atari, y lanzó la versión comercial de Tennis for Two, a la cual bautizó

## >PROGRAMACIÓN VIDEOJUEGOS (I)-



*Dos clásicos frente a frente: Gálaga, el clásico de los 80's que atrapó a casi todos los fanáticos, VS. Duke Nukem Forever, la resurrección del clásico FPS que trae las últimas novedades en gráficos.*

como PONG. La misma venía con una consola y dos joysticks, esta se conectaba al televisor y permitía pasar horas y horas de entretenimiento continuo.

Luego llegaron otros clásicos como Computer Space, y algunos años más tarde los ya conocidos Pac-

Man, Elevator y demás videojuegos que trazaron un camino de fama, dinero y fortuna a unos cuantos personajes del mundo computacional.

### **LA REALIDAD DE HOY**

Si bien repasar todo el espectro de evolución del videojuego podría requerir de un libro entero, hoy todos conocemos que la realidad de aquellas pantallas blanco y negro o de monitores de fósforo verde ha cambiado y mucho. La gran revolución del procesamiento y almacenamiento que venimos viviendo en las últimas tres décadas hacen que podamos ser partícipes en primera persona de videojuegos cada vez más reales en cuanto a calidad grá-

*1945, un videojuego gratuito que emula a los clásicos 1942 y 1943, desarrollado íntegramente en Game Maker.*



fica y respuesta de acciones. Esta industria ha reportado el último año ganancias millonarias a las empresas que aprovechan la cresta de la ola creada por el entretenimiento colectivo-social. Gracias al gran condimento aportado a la globalización por la gran red de redes, las aventuras gráficas que se van a lanzar al mercado en un futuro cercano, son inmediatamente conocidas por todos alrededor del mundo y esto, a su vez, permite sacar estadísticas bastantes más acertadas en las proyecciones de ventas.

### INICIARSE EN EL MUNDO DE LOS VIDEOJUEGOS

Muchos son los programas informáticos que nos permiten desarrollar juegos, y dependiendo de la complejidad de la aventura gráfica, podemos llegar a tardar desde una semana, para algo simple como una batalla espacial de gráficos sencillos, hasta algo más de un año para un juego del estilo First Person Shooter, y por supuesto que para este último requeriremos de un equipo de profesionales preparados para cubrir aspectos de lineamiento histórico, gráfica, arte, programación, testing y Marketing, entre otros más.

Existe también un mercado que ha crecido en la última década de manera considerable, que es el de los Indie Game Developers. Como su nombre lo indica, es un mercado que tiene pocas personas que conforman una sociedad o



empresa, y que se dedican a hacer videojuegos independientes. El mismo puede estar conformado por un programador, un diseñador y un entendido en el mundo del Marketing, que en conjunto diseñan una idea, la cual vuelcan en un videojuego, lo preparan y lo lanzan a la gran red de redes, haciendo publicidad viral en redes sociales, Twitter y algunos foros o sitios dedicados a esta temática. Así sus videojuegos van creciendo en popularidad y con el tiempo estas personas terminan trabajando de manera freelance para grandes corporaciones. Las herramientas utilizadas pueden ser muchas, o pocas, dependiendo de la complejidad del videojuego.

Una herramienta que cobró fuerza, un poco por accidente y otro poco por el apoyo incondicional y la motivación que crearon quienes la conocieron desde sus inicios fue Game Maker. Mark Overmars, su creador,

El IDE de Game Maker, simple e intuitivo al igual que cualquier aplicación Windows.

es un profesor de programación de videojuegos, quien desarrolló esta herramienta en principio para explicar a sus alumnos lo referente a geometría, imágenes y entornos virtuales, los tres principios fundamentales de todo videojuego. Este profesor de la Universidad de Utrecht, países bajos, volcó sus conocimientos de programación, en principio creando un software gráfico, al cual dado su éxito, terminó integrándole manejo multimedia y programación con su propio lenguaje, conocido como GML.

Actualmente Game Maker se encuentra en su versión 8, la cual analizaremos en esta nota, y sigue

## >PROGRAMACIÓN VIDEOJUEGOS (I)-



IMAGEN	NOMBRE DEL SPRITE
Planet1.png	Planet1
Planet2.png	Planet2
vulture_droid_b_1	Enemy_1



*Una herramienta que cobró fuerza, un poco por accidente y otro poco por el apoyo incondicional y la motivación que crearon quienes la conocieron desde sus inicios fue Game Maker.”*



avanzando a la par del mundo informático, prometiendo futuras mejoras para sus versiones venideras.

### **GAME MAKER, EL IDE**

Actualmente Game Maker posee dos versiones: la gratuita, para desarrollos independientes no comerciales de videojuegos en 2D, y una versión paga que permite desarrollar videojuegos en 2 y 3 dimensiones, y que integra a fondo su lenguaje de programación GML.

Su IDE es simple y fácil de entender, aunque nunca antes hayan programado videojuegos. Posee un menú donde agrupa las principales funciones, una barra de herramientas, donde resume un acceso rápido

a las funciones que utilizamos más a menudo, y una barra lateral con una serie de carpetas donde se irán anidando los distintos objetos con los que crearemos nuestro proyecto. La barra lateral contiene los siguientes elementos:

**Sprites:** Los sprites son los gráficos estáticos o animados que compondrán el videojuego.

**Sounds:** Aquí se agrupan los sonidos que darán vida auditiva a nuestra aplicación: música de fondo, disparos, explosiones, etc.

**Backgrounds:** Aquí se guardan las imágenes que componen el fondo de la aplicación, o área de desarrollo del videojuego.

**Paths:** Aquí se guardan los distintos trazos o ruta que le daremos a los recursos animados de nuestro juego.

**Scripts:** Contiene los bloques de código con los que se animarán los objetos del juego.

**Fonts:** Permite cargar fuentes personalizadas para los textos en pantalla de la aplicación.

**Time Lines:** Para los juegos que recorren una línea de tiempo, se creará un timeline.

La opción más fácil de agregar recursos al videojuego es a través del menú contextual que Game Maker nos pone en cada uno de los recursos.

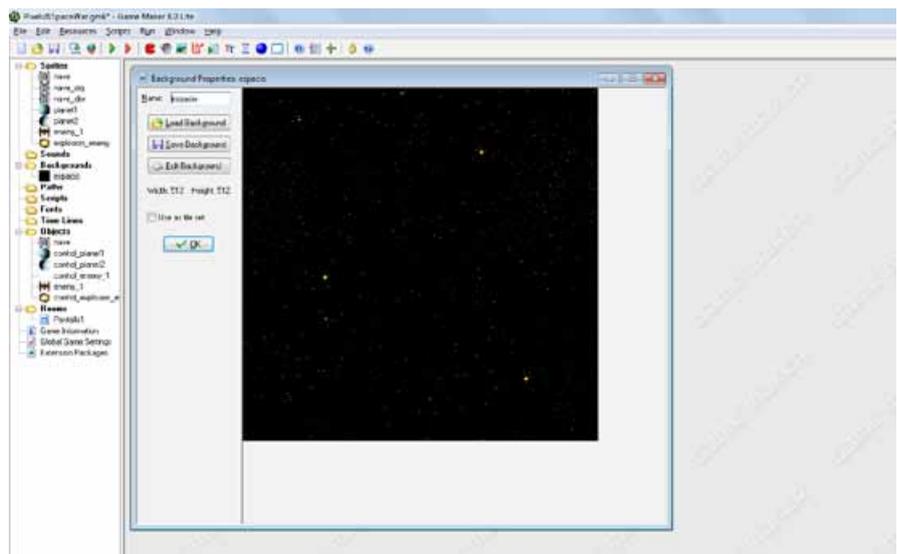
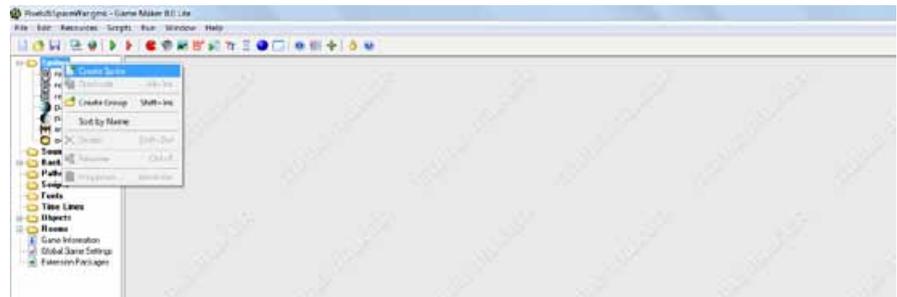
**Objects:** Aquí se crean los objetos basándose en los sprites y acciones que tendrá el videojuego, y se les dará la animación y coordenadas en pantalla.

**Rooms:** En esta carpeta se representará cada escenario del videojuego.

## NUESTRO PRIMER VIDEOJUEGO

Descarguemos e instalemos GameMaker desde el siguiente link: [bit.ly/pZG0Gg](http://bit.ly/pZG0Gg), instalemos la aplicación e iniciemos la misma desde los programas de Windows. Luego descarguemos el siguiente Pack de recursos de la dirección: <http://bit.ly/nIbC6V>. Creamos una carpeta llamada Pixels&Space y dentro de ella la carpeta Resources, donde copiamos el contenido del archivo RAR descargado. Para entender mejor a Game Maker, crearemos un videojuego del tipo Scrolling. Desde su IDE presionamos CTRL + N para crear un nuevo juego. Lo llamamos Pixels&Space y guardamos el proyecto en la carpeta creada.

El videojuego lo dividiremos en dos partes: en esta primera entrega crearemos el escenario principal, los objetos, algunas animaciones y pondremos a funcionar el videojuego. En la segunda entrega veremos cómo agregar dificultad, de qué manera podemos cruzar fuego contra



los malos y cómo contabilizar las vidas. Por último configuraremos y compilaremos el videojuego y le agregaremos una imagen inicial a la portada del mismo.

## CREANDO SPRITES

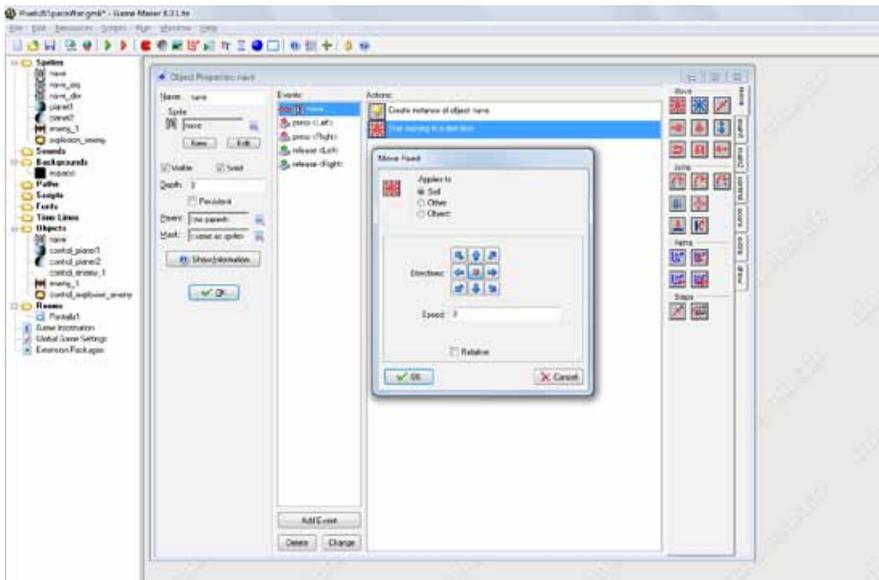
Agregamos en primera instancia los sprites, que serán los recursos que veremos en el videojuego. Sobre el menú lateral, en Sprites, presionamos el botón derecho del mouse y seleccionamos Create Sprite. Como nombre le pondremos Nave, y presionando Load Sprite, agregamos la imagen ship1.png, guardando el Sprite como último paso. Repetimos lo mismo, agregando la imagen ship1\_left.png, guardando

Utilizando una imagen simple como es el reflejo de las estrellas en el espacio, lograremos crear la sensación de desplazamiento continua con un simple wallpaper.

el sprite como Nave\_izq, y luego agregamos Sship1\_right.png, guardando el Sprite como Nave\_der. Realizamos lo mismo para las siguientes imágenes:

Con la salvedad de la última imagen, los pasos son repetitivos. Para Enemy\_1, agregamos la imagen vul-

## >PROGRAMACIÓN VIDEOJUEGOS (I)-



Como podemos apreciar, la edición de los eventos y acciones es 100% visual, lo que nos ayuda a comprender fácilmente qué hará nuestro videojuego.



*El videojuego lo dividiremos en dos partes: en esta primera entrega crearemos el escenario principal, los objetos, algunas animaciones y pondremos a funcionar el videojuego. En la segunda entrega veremos cómo agregar dificultad, de qué manera podemos cruzar fuego contra los malos y cómo contabilizar las vidas.”*



ture\_droid\_b\_1.png, luego presionamos Edit Sprite, y en la ventana Sprite Editor, seleccionamos de la barra de herramientas la opción Add a Sprite from a file. Obtendremos de esta manera una imagen animada. Pueden probarla tildando la opción lateral Show Preview.

### **CREANDO EL BACKGROUND**

A continuación agregaremos el área donde se llevará a cabo el juego. Tomamos el wallpaper background.png dentro de la carpeta Resources, y lo agregamos en la carpeta Backgrounds, presionando nuevamente el botón derecho del mouse sobre ésta. Guardamos el Background con el nombre Espacio.

### **TRANSFORMANDO IMÁGENES EN OBJETOS**

Nos quedan dos tareas más antes de finalizar la primera entrega. En principio, debemos darle vida a los objetos que serán los encar-

gados de animar el videojuego. Para ello, sobre la carpeta Objects presionamos botón derecho y seleccionamos Create Object.

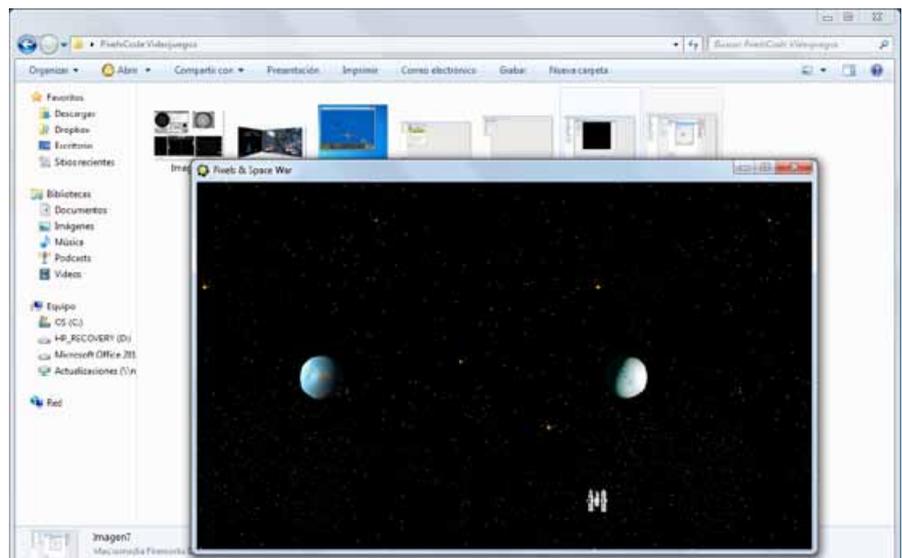
Como nombre le pondremos Nave. Seleccionamos el Sprite nave, tildamos Visible y Solid, y luego presionamos el botón Add Event. Seleccionamos Collision, luego agregamos las siguientes acciones sobre éste: Create Instance, y Start Moving in a direction, seleccionando para esta última la opción Center. Luego agregamos el evento Press <left>, y las acciones serán Starting moving in a direction (Izquierda), Speed = 7. El evento hará que al presionar la flecha izquierda, la nave se desplace hacia ese lugar de la pantalla. Luego agregamos la acción Change Sprite into, seleccionando el Sprite nave\_izq, Speed = 6. Agregamos a continuación el Evento Press <Right>, repitiendo las acciones anteriores, cambiando el Sprite por nave\_der, y la dirección (derecha). Luego agregamos los eventos Release <Left> y Release <Right>, los cuales se ejecutarán cuando soltemos las teclas. Agregamos

los eventos Starting moving in a direction (seleccionando Center), y Change Sprite into, seleccionando nave. De esta forma, nuestra nave tendrá movimientos hacia los laterales.

Por último agregamos dos Objects más que llamaremos Control\_planet1 y Control\_planet2, agregando como Sprites a Planet1 y Planet2 respectivamente. En Control\_planet1 agregamos el evento Create, y las acciones Set Vertical Speed = 1, y Set Alarm 0 to 164. Luego el evento Step, y las acciones If y is larger than room\_height + 60. Jump to position (random(room\_width), -60). Set Alarm 0 to 164. Guardamos el Objeto, y en Control\_planet2, repetimos lo mismo que en éste, cambiando lo siguiente: Set vertical Speed = 2. Set Direction and speed of motion (-81; 2), y Set Alarm 0 to 225.

### **ARMAR Y TESTEAR EL JUEGO**

Por último, antes de ejecutar nuestro videojuego para mirar su aspecto, sobre la carpeta Rooms presionamos el botón derecho del mouse, luego Add Room. Como nombre irá Pantalla1. En la pestaña objects seleccionamos Nave, y hacemos un clic sobre el área Room. Se agregará de esta manera nuestra nave al escenario. Seleccionamos a continuación Control\_planet1 y lo agregamos en algún lugar del Room. Lo mismo para Control\_planet2. Luego seleccionamos la pestaña Settings, donde pondremos en Caption of the room: Pixels & Space



War, Width: 800, Height: 480, Speed: 30. En la pestaña Tiles, seleccionamos espacio. En la pestaña Backgrounds seleccionamos Color Black, Background 0, tildamos Visible when the room starts y Foreground image: espacio. Guardamos todas las ventanas y presionamos el botón PLAY, de color verde de la barra de herramientas.

Nuestro juego se ejecutará, y podremos ver la animación de los planetas desplazándose por la pantalla y el efecto de desplazamiento creado sobre el Wallpaper espacio. Presionando las flechas Izquierda - Derecha del teclado moveremos nuestra nave, con los efectos aplicados sobre la misma. Si presionamos F4 el juego se verá en pantalla completa. De la misma forma restauramos el juego al tamaño especificado en el Room. Presionando la tecla ESC,

*El videojuego en Acción: simplemente podremos mover nuestra nave con las flechas del teclado, pero también podremos apreciar los efectos de desplazamiento del espacio y los planetas 1 y 2 agregados.*

saldremos del mismo. La próxima entrega tendrá enemigos, disparos y explosiones por doquier, lo que hará al juego más emocionante y funcional. **P**



# Introducción a las bases de datos relacionales (VI)

Por Juan Gutmann .....  
juan.gutmann@pixelscode.com

*Continuamos avanzando con las prácticas empleando el motor PostgreSQL. En la quinta entrega de esta serie de notas comenzamos a trabajar con Stored Functions, el equivalente en PostGres a los populares Stored Procedures, una solución que trae múltiples beneficios a los sistemas que la utilicen. Algunos de ellos son velocidad de ejecución, baja en el tráfico de la red y simplificación del codebase, así como una reducción significativa del impacto que produce un cambio en el modelo de datos o las reglas de negocio sobre las “aplicaciones cliente” que trabajen contra una base de datos relacional. En esta ocasión vamos a trabajar nuevamente con PL/pgSQL, el lenguaje programático de PostGres inspirado en el PL/SQL de Oracle, pero incorporando validaciones mediante el empleo de Triggers, y nuevos elementos del lenguaje, como sentencias de control de flujo y cortes de control.*

## **EL PROBLEMA A RESOLVER**

Supongamos que a nuestra base de práctica, en la que ya tenemos almacenados los datos del personal de una empresa, se desea agregar la información sobre los períodos de licencia de los empleados. El analista encargado de modelar la solución lo resuelve con una sola entidad, que bautiza como “licencia”, a la que dota de tres atributos: legajo, fecha\_desde y fecha\_hasta. Para garantizar la unicidad de los registros, los tres campos deben formar parte de la clave primaria. El código SQL necesario para crear la tabla es el siguiente:

```
CREATE TABLE licencia (  
legajo          NUMERIC(5) NOT NULL,  
fecha_desde     DATE       NOT NULL,  
fecha_hasta     DATE       NOT NULL,  
CONSTRAINT pk_licencia PRIMARY KEY (legajo, fecha_desde, fecha_hasta)  
);
```

Una vez creada la tabla, ingresamos algunos registros de prueba, que nos permitirán posteriormente testear los procedimientos que trabajen con ella.

```
INSERT INTO licencia  
(legajo, fecha_desde, fecha_hasta)  
VALUES  
(2, TO_DATE('03/01/2011', 'dd/mm/yyyy'), TO_DATE('16/01/2011', 'dd/mm/yyyy'));
```

```
INSERT INTO licencia  
(legajo, fecha_desde, fecha_hasta)  
VALUES  
(2, TO_DATE('23/05/2011', 'dd/mm/yyyy'), TO_DATE('29/05/2011', 'dd/mm/yyyy'));
```

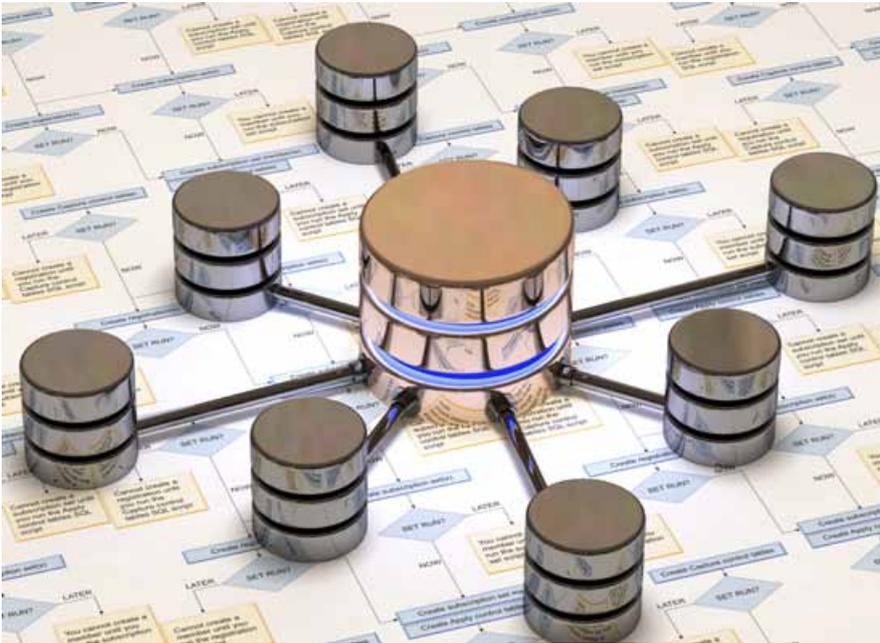
Naturalmente, al momento de ingresar datos en la misma, se deben validar la consistencia de los mismos: que para cada licencia la “fecha hasta” sea mayor o igual a la “fecha desde”, y que los períodos de licencia para un mismo empleado no se superpongan. Existe una forma de realizar estas validaciones en el mismo motor relacional, sin depender de que las realicen las aplicaciones cliente. Veamos cómo hacerlo.

## USANDO TRIGGERS

Los Triggers (en castellano, “disparadores”) son una forma de asociar un procedimiento almacenado a un evento (inserción, modificación o borrado de uno o más registros) sobre una tabla en particular. En PostGres, debemos escribir en primer lugar la Stored Function que debe llamarse cuando tiene lugar dicho evento, y posteriormente crear el trigger que se encargará de ejecutarla cuando esto ocurra. Una función asociada a un trigger tiene ciertas características especiales. No posee argumentos de entrada, y el argumento de salida debe ser de tipo “trigger”. Cuenta con ciertas estructuras internas que nos permiten referirnos a los valores nuevos o viejos del registro. Cuando ocurre un INSERT o UPDATE, podemos usar NEW para acceder a los valores nuevos, mientras que en un trigger de UPDATE o DELETE podemos referirnos a los valores previos a la modificación con la estructura OLD. Empleando PL/pgSQL, programamos la función para que valide que el rango de fechas al insertar un nuevo registro en “licencia” sea consistente. Para ello, la fecha desde debe ser menor o igual a la fecha hasta.

```
CREATE FUNCTION f_trg_licencia_ins() RETURNS trigger AS $$  
BEGIN  
    IF NEW.fecha_desde > NEW.fecha_hasta THEN  
        RAISE EXCEPTION 'Rango de Fechas Incorrecto, por favor verifique.';  
    END IF;  
  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

## >BASES DE DATOS (I)-



Los Triggers (en castellano, “disparadores”) son una forma de asociar un procedimiento almacenado a un evento (inserción, modificación o borrado de uno o más registros) sobre una tabla en particular.”

Ahora creamos el trigger propiamente dicho, indicando en qué momento debe ejecutarse la función que acabamos de desarrollar. En este caso, lo buscado es que lo haga para cada registro nuevo que se cree en la tabla “licencia”:

```
CREATE TRIGGER trg_licencia_ins
  BEFORE INSERT ON licencia
  FOR EACH ROW
  EXECUTE PROCEDURE f_trg_licencia_ins();
```

Para probar la validación, probamos insertar un registro que no cumple con la condición de validación.

```
INSERT INTO licencia
(legajo, fecha_desde, fecha_hasta)
VALUES
(2, TO_DATE('10/09/2011', 'dd/mm/yyyy'), TO_DATE('05/09/2011', 'dd/mm/yyyy'));
```

La respuesta de la base es un mensaje de error, concretamente el que hemos programado en la función para este caso:

```
ERROR: Rango de Fechas Incorrecto, por favor verifique.
```

Si hacemos un query sobre la tabla, confirmamos que el registro que incumple la validación no ha sido agregado.

```
SELECT * FROM licencia;

legajo | fecha_desde | fecha_hasta
-----+-----+-----
      2 | 2011-01-03 | 2011-01-16
      2 | 2011-05-23 | 2011-05-29
```

La otra condición que debemos comprobar es que los períodos de licencia para un mismo empleado no se superpongan. Para ello, agregamos unas pocas líneas a la función de validación.

```
CREATE OR REPLACE FUNCTION f_trg_licencia_ins() RETURNS trigger AS $$
DECLARE
    v_count NUMERIC;
BEGIN
    IF NEW.fecha_desde > NEW.fecha_hasta THEN
        RAISE EXCEPTION 'Rango de Fechas Incorrecto, por favor verifique.';
    END IF;

    SELECT COUNT(*)
    INTO v_count
    FROM licencia
    WHERE legajo = NEW.legajo
        AND (fecha_desde, fecha_hasta) OVERLAPS
            (NEW.fecha_desde, NEW.fecha_hasta);

    IF v_count > 0 THEN
        RAISE EXCEPTION 'Este período de licencia se superpone con uno ya
existente, por favor verifique.';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Una vez compilada esta nueva versión de la función de validación, comprobamos nuevamente que la condición agregada funcione bien con un INSERT.

```
INSERT INTO licencia
(legajo, fecha_desde, fecha_hasta)
VALUES
(2, TO_DATE('05/01/2011', 'dd/mm/yyyy'), TO_DATE('10/01/2011', 'dd/mm/yyyy'));
```

PostgreSQL responde:

```
ERROR: Este período de licencia se superpone con uno ya existente, por favor ve-
rifique.
```

Los siguientes registros también se superponen con períodos ya existentes y serán rechazados por el trigger:

```
INSERT INTO licencia
(legajo, fecha_desde, fecha_hasta)
VALUES
(2, TO_DATE('05/01/2011', 'dd/mm/yyyy'), TO_DATE('20/01/2011', 'dd/mm/yyyy'));
```

```
INSERT INTO licencia
(legajo, fecha_desde, fecha_hasta)
VALUES
(2, TO_DATE('01/01/2011', 'dd/mm/yyyy'), TO_DATE('20/01/2011', 'dd/mm/yyyy'));
```

En cambio, este otro caso cumple perfectamente con lo establecido y será agregado a la tabla sin inconvenientes.

```
INSERT INTO licencia
```

## >BASES DE DATOS (I)-

```
(legajo, fecha_desde, fecha_hasta)
VALUES
(2, TO_DATE('18/01/2011', 'dd/mm/yyyy'), TO_DATE('18/01/2011', 'dd/mm/yyyy'));
```

### **CONTANDO DÍAS HÁBILES**

Ahora vamos a concentrarnos en el problema principal: supongamos que una vez creada la tabla de acuerdo al diseño del analista, se vuelcan los datos de las licencias de todo el personal. Posteriormente, surge la necesidad de establecer la cantidad total de días de licencia que se ha tomado cada empleado. Aquí ya surge una dificultad inicial: solo deben considerarse como días de licencia los días hábiles, es decir, no deben contabilizarse los sábados, domingos ni feriados. Es fácil saber si una fecha determinada cae en día sábado o domingo. La forma más sencilla probablemente sea recurrir a las máscaras de formato que se utilizan con la función de conversión TO\_CHAR(), capaz de transformar un valor de tipo fecha a una cadena de caracteres (STRING). Si a un valor de fecha le aplicamos la función TO\_CHAR(fecha, 'D') obtendremos como respuesta una cadena conteniendo el número de día de la semana: 1 para Domingo, 2 para Lunes, 3 para Martes y así sucesivamente. Por ende, recurriendo a este método, todas las fechas en las que esta conversión no devuelva '1' y '7' (sábado) son hábiles. Pero todavía debemos considerar los días feriados. Aquí la cosa se complica, ya que distintas regiones de nuestro globo tienen distintas fechas festivas, y en todos los casos hay algunas, como Semana Santa, que no caen todos los años en la misma fecha. Lógicamente, la única solución posible es crear una tabla para almacenar las fechas que son feriado en cada año, y cuando necesitemos saber si un día es feriado o no, realizaremos una búsqueda sobre esta tabla. Si la fecha buscada no figura en ella, entonces definitivamente se trata de un día hábil. Creamos entonces la tabla "feriado" y le ingresamos algunos registros:

```
CREATE TABLE feriado (
fecha_feriado DATE NOT NULL,
descrip_feriado VARCHAR(100) NOT NULL,
CONSTRAINT pk_feriado PRIMARY KEY (fecha_feriado)
);
```

```
INSERT INTO feriado
(fecha_feriado, descrip_feriado)
VALUES
(TO_DATE('01/01/2011', 'dd/mm/yyyy'), 'Año Nuevo');
```

```
INSERT INTO feriado
(fecha_feriado, descrip_feriado)
VALUES
(TO_DATE('25/05/2011', 'dd/mm/yyyy'), 'Aniversario de la Revolución de Mayo');
```

```
INSERT INTO feriado
(fecha_feriado, descrip_feriado)
VALUES
(TO_DATE('09/07/2011', 'dd/mm/yyyy'), 'Día de la Independencia');
```

### **IMPLEMENTANDO LA SOLUCIÓN**

Para llevar a cabo lo solicitado, decidimos escribir dos Stored Functions separadas. Modularizar la solución de un problema, descomponiéndola en dos o más programas que resuelvan individualmente tareas más sencillas casi siempre es buena idea, ya que favorece tanto la lectura y comprensión de los programas, como su posterior reutilización. En primer lugar, escribimos una función que recibe como parámetros "fecha desde" y "fecha hasta", y retorna la cantidad de días hábiles comprendidos en el período delimitado por ambas fechas. Para ello, se tienen en cuenta los dos criterios anteriormente detallados: si un día cae en sábado, en domingo, o se encuentra presente

en la tabla de feriados, no se contabiliza como día hábil.

```

CREATE OR REPLACE
FUNCTION f_dias_habiles(
  adt_fecha_desde licen-
cia.fecha_desde%TYPE,
  adt_fecha_hasta
licencia.fecha_hasta%TYPE
)
  RETURNS NUMERIC AS $$
DECLARE
  v_fecha      DATE;
  v_feriado    NUMERIC;
  v_dias       NUMERIC :=
0;
BEGIN
  v_fecha := adt_fecha
desde;

  WHILE v_fecha <= adt
fecha_hasta LOOP
    IF TO_CHAR(v_fecha, 'D') IN ('1', '7') THEN
      v_fecha := v_fecha + 1;
      CONTINUE;
    END IF;

    SELECT COUNT(*)
    INTO v_feriado
    FROM feriado
    WHERE fecha_feriado = v_fecha;

    IF v_feriado > 0 THEN
      v_fecha := v_fecha + 1;
      CONTINUE;
    END IF;

    v_dias := v_dias + 1;
    v_fecha := v_fecha + 1;
  END LOOP;

  RETURN v_dias;
END;
$$ LANGUAGE 'plpgsql';

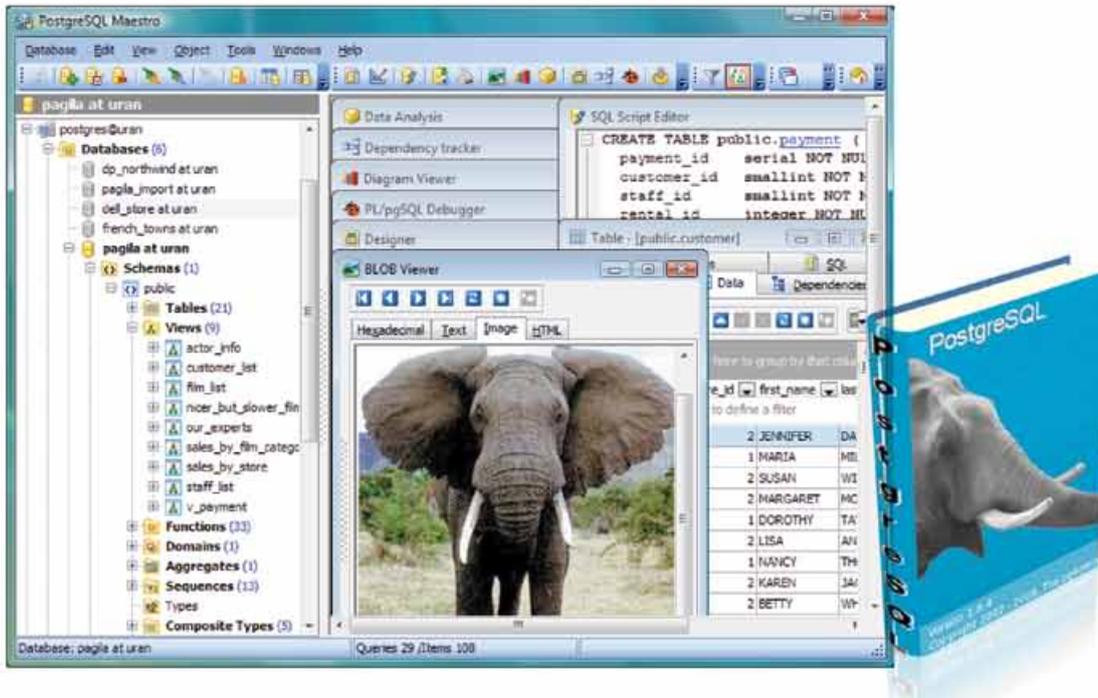
```

FECHA_FERIADO	DESCRIP_FERIADO
82 01/01/2007	Año nuevo
83 24/03/2007	Día de la memoria
84 02/04/2007	Día del veterano
85 05/04/2007	Jueves Santo
86 06/04/2007	viernes santo
87 01/05/2007	Día del trabajador
88 25/05/2007	Revolución de mayo
89 18/06/2007	MANUEL BELGRANO
90 09/07/2007	Día de la independencia
91 20/08/2007	Día del libertador
92 15/10/2007	Día de la Raza
93 08/12/2007	Día de la virgen
94 24/12/2007	Nochebuena
95 25/12/2007	Navidad
96 31/12/2007	Vísperas de Año Nuevo
97 01/01/2008	Año Nuevo
98 20/03/2008	Jueves Santo
99 21/03/2008	Viernes Santo
100 24/03/2008	Día de la Memoria
101 02/04/2008	Día de la Veterano de Malvinas
102 01/05/2008	Día del Trabajador
103 25/05/2008	Primer Gobierno Patrio
104 16/06/2008	Día de la Bandera
105 09/07/2008	Día de la Independencia
106 18/08/2008	Día del Libertador
107 13/10/2008	Día de la Raza

“Modularizar la solución de un problema, descomponiéndola en dos o más programas que resuelvan individualmente tareas más sencillas casi siempre es buena idea, ya que favorece tanto la lectura y comprensión de los programas, como su posterior reutilización.”

Analicemos el funcionamiento de este ejemplo. Comenzamos asignando a la variable `v_fecha` el valor del argumento `fecha_desde`. Mediante las sentencias de control de flujo `WHILE ... LOOP` vamos a iterar repetidamente un bloque de código, hasta que se cumpla la condición establecida: que la variable `v_fecha` sea menor o igual al argumento `fecha_hasta`. Dentro de ese bloque, para cada valor de `v_fecha` (es decir, para cada día comprendido dentro del

## >BASES DE DATOS (I)-



período fecha\_desde .. fecha\_hasta) primero nos fijamos si ese día es un Sábado o Domingo. En caso afirmativo, le sumamos un día a v\_fecha y pasamos a la siguiente iteración con un “CONTINUE”. Esta sentencia suspende la ejecución del bloque de código que se encuentra dentro del loop para la iteración actual y continúa con la siguiente iteración, siempre y cuando se mantenga la condición establecida en el WHILE. Luego buscamos si el valor actual de v\_fecha se encuentra en la tabla de feriados. Igual que con los fines de semana, en caso afirmativo incrementa el valor de v\_fecha y salta a la siguiente iteración. Si el flujo del código llegó hasta aquí, quiere decir que la fecha contenida en v\_fecha es efectivamente un día hábil, por lo que se le suma uno a la variable acumuladora v\_dias, e incrementa v\_fecha. Una vez terminado el loop, tendremos en v\_dias la cantidad de días hábiles en el período analizado, por lo que finalizamos retornando ese valor. Luego de compilada la función, para probar su correcto funcionamiento, teniendo en cuenta que uno de los valores de prueba insertados en la tabla de feriados es el 25 de Mayo, ejecutamos el siguiente query:

```
SELECT f_dias_habiles(TO_DATE('23/05/2011', 'dd/mm/yyyy'),  
                     TO_DATE('29/05/2011', 'dd/mm/yyyy'));
```

La respuesta del motor PostGres es 4, indicando que todo funciona como es debido.

Para completar lo requerido (obtener en base a la tabla existente la cantidad total de días de licencia de un empleado determinado) escribimos otra Stored Function que se valga de la anterior para obtener el valor buscado:

```
CREATE OR REPLACE FUNCTION f_dias_licencia(  
    an_legajo      licencia.legajo%TYPE )  
    RETURNS NUMERIC AS $$  
DECLARE  
    v_dias  NUMERIC;  
BEGIN  
    SELECT SUM(f_dias_habiles(fecha_desde, fecha_hasta))  
    INTO v_dias
```

```

FROM licencia
WHERE legajo = an_legajo;

RETURN v_dias;
END;
$$ LANGUAGE 'plpgsql';

```

Consiste en un simple query SELECT que obtiene para el empleado que recibe como parámetro todos los períodos (registros) existentes en la tabla de licencias, invocando para cada uno de ellos la función f\_dias\_habiles() que escribimos antes, y sumando todos los valores retornados por ésta con SUM(). De hecho, esta última Stored Function es tan sencilla que podríamos escribirla directamente en SQL puro, en lugar de emplear PL/pgSQL, de la siguiente forma:

```

CREATE OR REPLACE FUNCTION f_dias_licencia(
  an_legajo licencia.legajo%TYPE )
RETURNS NUMERIC AS $$
SELECT SUM(f_dias_habiles( fecha_desde, fecha_hasta))
FROM licencia
WHERE legajo = $1;
$$ LANGUAGE 'sql';

```

La única “pega” de esta versión, es que al escribirla en SQL puro, en el cuerpo de la función no podemos hacer referencia a los argumentos (parámetros) por su nombre, sino que tenemos que hacerlo por su posición. Al primer argumento nos referimos como \$1, si hubiera un segundo sería \$2, y así sucesivamente. De todas formas, muchos programadores prefieren trabajar siempre directamente en PL/pgSQL en este tipo de casos, ya que si se requieren posteriores modificaciones a la función que necesiten más lógica, o algún tipo de control de flujo, habrá que trabajar menos para aplicarlas.

Una vez compilada una de las dos versiones de f\_dias\_licencia, procedemos a testearla con este query:

```

SELECT f_dias_licencia(2);

```

La base responde “14”, lo cual es correcto, ya que el empleado cuyo legajo es 2 cuenta con dos períodos de licencia: uno en el mes de enero, de 10 días hábiles, y otro en mayo, de 4 días.

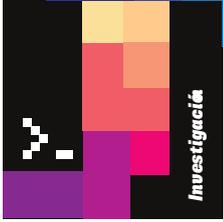
### **... Y HASTA AQUÍ LLEGAMOS EN ESTA OCASIÓN**

Continuamos profundizando en las posibilidades del código almacenado en el motor relacional. A medida que avanzamos, quedan cada vez más claras las ventajas de su empleo. Además de las ya mencionadas previamente, al realizar estas prácticas también apreciamos que permite programar en poco tiempo reglas complejas de negocio, en un lenguaje con sintaxis clara y sencilla, proveniente del legendario ADA. Amerita destacar que, adicionalmente, contamos con el beneficio de que no nos exponemos a que una aplicación cliente pobremente codificada (por ejemplo, si no realiza las validaciones correspondientes) corrompa la integridad de la información almacenada en la base, un punto definitivamente vital para el buen funcionamiento de cualquier sistema que trabaje con un sistema de base de datos relacional. En el próximo número continuaremos con estas prácticas, incorporando nuevos elementos de PL/pgSQL, que nos permitirán escribir programas más sofisticados y poderosos. Como siempre, los invitamos a escribirnos para compartir con nosotros sus dudas, comentarios o consejos sobre este fascinante tópico. ¡Hasta la próxima! **P**



*La única “pega” de esta versión, es que al escribirla en SQL puro, en el cuerpo de la función no podemos hacer referencia a los argumentos (parámetros) por su nombre, sino que tenemos que hacerlo por su posición.”*





# e-Voto, el diseño del nuevo universo electoral

Por Roxana Miguel

roxana.miguel@pixelscode.com

@roxanamiguel



*El voto electrónico ha llegado a la sociedad de la información para agilizar el sistema tradicional y ofrecer mayor seguridad. Pese a que gradualmente se está instalando en el mundo, sigue siendo muy cuestionado. Desarrolladores y diseñadores concentran su trabajo en hacer que las decisiones electrónicas tengan validez. Aquí conoceremos las pautas generales para el armado de un software libre.*

**E**s la expresión ciudadana por medios electrónicos que, a diferencia del voto tradicional, ofrece más seguridad y eficacia. Si bien estos puntos son los beneficios que tiene a favor, también son los más cuestionados debido a que todos conocemos cuán inseguras pueden ser las transferencias de datos en una sociedad de la información donde la piratería y hackeo de cuentas electrónicas o bancarias se han convertido en temores cotidianos. El número de transferencias electrónicas aumenta progresivamente y esto se debe a que las personas eligen este medio no sólo para comprar productos y pagar servicios sino también para compartir información laboral y personal, participar de encuestas sobre productos que les interesan, etc. Aunque la seguridad es un tema aún indefinido, ya que crecen los métodos que ofrecen

confianza y al mismo tiempo crecen las formas de flanquearlos, es cada vez más evidente que muchas de nuestras actividades pasan por una computadora. Si cada vez son más las diligencias que pueden realizarse desde una computadora doméstica, de oficina y hasta desde un smartphone, ¿por qué no se podría emitir un sufragio electrónico también? Países como Alemania, Italia, Australia, Suiza, Bélgica, Canadá, Estonia, Francia, India, Irlanda, Holanda, Noruega, Rumania, Reino Unido, Brasil, Venezuela y Argentina ya lo han puesto a prueba y han experimentado al mismo tiempo sus controversias. Si bien al hablar de voto electrónico pensamos en el sufragio, el término hace referencia a distintos modos de expresión electrónica que, en definitiva, tienen que ver con la posibilidad de que una persona pueda



tomar una decisión y expresarla por medio de un servicio informático configurado para un fin específico: una encuesta en un diario online, un formulario con opciones de respuesta en un blog y los sondeos de opinión, también entran dentro del modo de voto electrónico para los que se utilizan diferentes tipos de software libre.

La decisión electoral electrónica entonces es una forma más de transferencia de datos. A lo largo de este artículo veremos que pensar un software de voto electrónico no es demasiado complejo.

Aunque para todos los casos lo que importa es que el resultado sea verídico y eficaz, convengamos que en un recuento electoral no pueden surgir errores, pues las consecuencias modificarían el principio básico de la democracia. Por lo tanto, al

diseñar un software de voto electrónico conviene que tengamos presente la legislación de sufragio y de tratamientos de datos personales, aún cuando algunos puntos podamos tomarlos como evidentes. Una vez puestos a estudiar las formas de e-voto tenemos que saber que existen distintas tecnologías, pudiendo ser mediante: tarjetas perforadas, escáner óptico, grabación electrónica directa (DRE), vía telefónica, computadoras especiales o Internet. El diseño del programa dependerá de las herramientas con las que se brindará el servicio, pero en definitiva todos cuentan con un planteo general basado en contadores de datos.

### **¿CÓMO DEBIERA SER UN PROGRAMA E-VOTO?**

Resulta que es de lo más sencillo: por un lado tenemos una lista de votantes, por el otro una con las

opciones y finalmente un contador. Dos funcionan como arrays y el otro como un modo de recuento. Así logramos definir el objetivo del voto electrónico: conocer los resultados de lo que la gente elige de entre una esfera de candidatos/opciones.

Al tratarse del sufragio, que es un derecho político y constitucional, tendremos que tener en cuenta algunas cuantas consideraciones para que este sea electrónico. El software debe ofrecer parámetros que garanticen los resultados y que estos respondan al mandato de la democracia: universal, libre, directo, igualitario y secreto. Veremos a continuación que muchos programadores prefieren software libre, por ofrece más transferencia para estos fines.

Estos parámetros incluyen garantizar un ligamiento con los protocolos legales, funcionar en cualquier plataforma o sistema, con capacidades gráficas y sonoras que lo hagan accesible para personas con cualquier tipo de impedimento físico o mental, debe proveer de información para el votante como foros de debate y correo electrónico, además de garantizar la seguridad, individualidad y privacidad del voto.

Para el armado de este software entonces contaremos con una base de datos, computadoras o máquinas de voto, un software que pueda exponer visiblemente el objetivo del mismo y programas libres que compatibilicen con el software para que la información sea transferida de forma segura y sin interferencias. Una vez que tenemos una base de

### CLASIFICACIÓN DE DISTINTOS SISTEMAS:

**Sistema de toma de decisiones:** Normalmente son aplicaciones que acompañan programas de videoconferencia o trabajo cooperativo. Un ejemplo de ello es *Vote-Debian*, un sistema basado en el uso del correo electrónico donde, mediante una propuesta se genera un debate, una votación y una emisión por correo de los resultados. Está orientado a organizar el trabajo de un grupo de desarrolladores de Debian. (<http://vote.debian.org>)

**Sistemas de recuento de votos:** Los más populares son el mecanismo de kiosco donde autenticado el votante se autoriza su voto en una cabina. El recuento se hace de forma automática al final de la jornada. Otro, es el lector óptico donde el votante coloca en una urna o dispositivo con lector óptico su voto electrónico. Éste cuenta con un código de barras o chip que contiene la información.

**Sistema de encuestas o consultas anónimas:** Portales como *Slashdot* o *BarraPunto* ofrecen foros de discusión con temas que provee el administrador. La gente participa por medio de encuestas o en debates públicos dando su opinión.

**Sistemas de voto:** Son programas simples, basados en php y MySQL en su mayoría, que cuentan con un fichero que contiene la consulta y opciones de respuesta. En la mayoría de los casos los resultados se conservan mediante cookies. Un ejemplo de ello es *MyPools*, un sistema desarrollado sin posibilidades de reconfiguración y donde el administrador procede de manera manual en la mayoría de las operaciones.



**El programa debe garantizar que el voto sea único y personal, autenticando que la identidad del votante sea la real.”**

datos, un sistema de estándares abiertos y otro de comunicación entre el votante y, en este caso la junta electoral, es momento de definir los inconvenientes con que nos encontraremos al diseñar el programa.

- Problemas de autenticación: El programa debe garantizar que el voto sea único y personal, autenticando que la identidad del votante sea la real. Por lo que se debe certificar la máquina desde la que vota, el o los servidores del sistema de votación, la identidad del votante

y el sistema de voto en sí mismo. Normalmente se recurre al uso de cookies para identificar y hacer un seguimiento del usuario, pero en este caso no es recomendable puesto que estos pueden modificarse y hasta falsificarse sin control del servidor. En su defecto, se puede implementar un sistema de claves que autoricen la entrada al sistema y otra para la emisión de voto, o bien generar un protocolo de encriptado con PGP con agentes de certificación para ambos niveles: en el ingreso al sistema y al momento de votar.

- Problemas de visibilidad de los resultados: Si bien vemos que el método de encriptación puede resultar una opción muy buena, no todos los sistemas de votos electrónicos cuentan con ello y es aquí donde nos topamos con uno de esos parámetros que la ley nos exige: el voto debe ser secreto. Esto debe ser así tanto al momento de sufragar, al hacer el recuento y también cuando se dan los primeros datos parciales. Esto dependerá de cómo esquematicemos este punto al generar una fragmentación de la información. Conviene plantearnos mensajes con el menor número de opciones a contabilizar en un solo contador y cuyos resultados se evalúan al mismo tiempo que se generan, de manera que el sistema sea totalmente flexible de exponer datos parciales sólo si fueran solicitados y sin poder individualizarlos.

- Problemas de seguridad: Al trabajar con una base de datos, nuestro programa puede ser debilitado por ataques externos, fallas del software o del equipo, incluso hasta manipulaciones personales del adminis-

trador. En este último caso la Ley 25.326 sobre el tratamiento de datos personales establece en su artículo 9 que “1. El responsable o usuario del archivo de datos debe adoptar las medidas técnicas y organizativas que resulten necesarias para garantizar la seguridad y confidencialidad de los datos personales, de modo de evitar su adulteración, pérdida, consulta o tratamiento no autorizado, y que permitan detectar desviaciones, intencionales o no, de información, ya sea que los riesgos provengan de la acción humana o del medio técnico utilizado”. Asimismo “2. Queda prohibido registrar datos personales en archivos, registros o bancos que no reúnan condiciones técnicas de integridad y seguridad».

Por tanto debemos considerar este punto como fundamental. Los problemas pueden ser desperfectos técnicos ocasionados por un corte de luz, hasta interferencias de externos en el sistema. Para el primer caso con tener baterías de repuesto es suficiente y en el segundo, un buen sistema de encriptado o chips de corto alcance, incluso hasta una ubicación continua de las máquinas emisoras del voto terminaría por generar demasiada interferencia como para que un sistema externo logre captar y leer los resultados.

- Problemas de verificación: ¿Cómo saber que la información introducida a una base de datos no terminará por perderse? Para que un programa de voto electrónico sea realmente eficaz debe ofrecer la posibilidad de verificar que el mismo cumple con el objetivo y no otros. Es así como una vez más vemos que el software libre es el medio ideal para sistemas

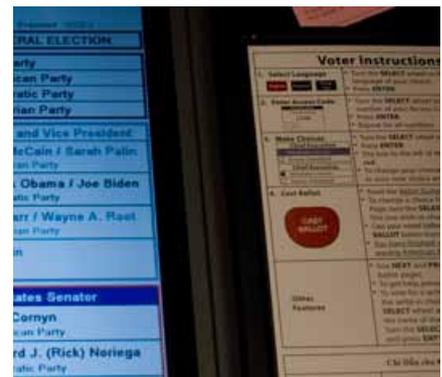
seguros. Al poder acceder al código de fuente y su compatibilidad con el servidor, entonces cualquiera, incluso un votante, podría verificar su funcionamiento.

Existen muchos casos de programas cuyas fallas han terminado con la ilusión de hacer de los sufragios un buen negocio.

### **UN UNIVERSO DE SISTEMAS INCOMPLETOS**

Por una razón que queda a criterio de cada uno debatir y como ya lo hemos mencionado antes, no todos los sistemas de voto electrónico desarrollados hasta el momento son totalmente completos, es decir, que cuenten con técnicas de certificación,

encriptación y verificación de datos. Es que todos estos requisitos no son estrictamente necesarios y terminan por generar una complicación a la hora de programar algo tan sencillo como un contador, sin



### **EL CASO DEL E-VOTO SALTEÑO**

*MSA ha desarrollado un software libre que se puso en práctica en la provincia de Salta en donde se fusionan la estructura física y lógica para aprovechar la tecnología, mediante una Boleta de Voto Electrónico (BVE) que está conformada por dos extremos troquelados.*

*El votante acude a la mesa en donde se retira un troquel con un código, se acerca a la máquina de votos y selecciona de un menú táctil las opciones para conformar su propia lista, una lista completa o emitir un voto en blanco. Finalizados estos pasos, se apoya el BVE en la impresora para que de un lado deje impresa la elección realizada.*

*Con esa boleta vuelve a la mesa, el fiscal confirma que el troquel restante coincida con el retirado y el votante deposita su boleta en la urna. La boleta cuenta con un chip que al momento de ser impreso se quema para no poder modificarlo y su alcance es tan corto que no podría ser detectado por ningún sistema de piratería.*

*Para su funcionamiento requiere:*

*Un ordenador (PC), con Pantalla Sensible al Tacto (Touch Screen)*

*Una impresora de BVE - Texto claro /Transferencia Térmica con placa Lectora y Grabadora de Chips de RFID*

*Un software de navegación para construcción del voto*

*Boletas de Voto Electrónico*

*Un dispositivo de Verificación electrónica de contenidos grabados en memoria digital*

*Un teclado numérico (Sistema Braille, auriculares, software para navegación auditiva y conformación de voto sobre BVE® para equipos especiales).*

*Un software de consolidación de datos e impresión de actas para los escrutinios, como así también un software de encriptación de datos y transmisión.*

*Se utiliza con sistema operativo Linux y la aplicación de votación se ejecuta desde un CD que contiene todo el software requerido.*

embargo, en el caso de diseñar un programa que tendrá la responsabilidad de transmitir la elección política de los ciudadanos lo correcto sería tomar todas las precauciones que están a nuestro alcance. A continuación veremos algunos

ejemplos de fallas en los sistemas de votos electrónicos:

El caso Diebold Election Systems es uno de los más emblemáticos, dado que en las elecciones de 2004 los condados de Alameda y San Diego, en Estados Unidos, fueron afecta-

dos por tarjetas de votación cuyos códigos no funcionaban. Pese a que el secretario de Estado incriminó penalmente y pidió el retiro total del servicio, y que además la compañía recibió una acusación legal por casos fraudulentos, esta pagó la demanda y en el 2006 volvió a la esfera electrónica bajo el nombre Premier Election Solutions, esta vez como un sistema de votación por escaneo óptico.

No menor es el caso de Nada, en California, donde en el mismo año, un escáner obvió el registro de más de seis mil quinientos votos por estar mal calibrado. Mientras que en 2006, en Holanda, el servicio de inteligencia pudo demostrar que se podían “escuchar” los votos de 1187 máquinas de votar mediante un dispositivo llamado Van Eck. Para entonces en aquel país ya existía una organización llamada “no confiamos en las computadoras de votación”.

También en California, pero en 2007, se solicitó un informe sobre la seguridad de las computadoras de votación, incluyendo la evaluación del código fuente del sistema para detectar vulnerabilidades, adulteraciones y errores en máquinas de las compañías Diebold Election Systems, Hart InterCivic, Sequoia Voting Systems y Elections Systems and Software, Inc. Como resultado se retiraron algunas, se detectaron que unas cuantas no cumplían requisitos mínimos de seguridad y que en otros casos eran tan vulnerables que podía instalarse software de robo de datos en menos de un minuto durante las elecciones.

En la Argentina, en el año 2007, la empresa Altec implementó el voto electrónico en la ciudad rionegrina

de Las Grutas. Dos de las urnas impidieron sufragar al 25% del padrón de mujeres y cuatro resultaron con cero votos. El municipio terminó por abolir el método. Mientras que en 2009, en Brasil, se contrataron 32 especialistas para probar la seguridad de las urnas electrónicas. Finalmente uno logró romper el secreto de sufragio con lectura de radiofrecuencia utilizando equipos de bajo costo.

### **CUESTIONES ESTRUCTURALES**

A modo de conclusión podemos decir que para que nuestro programa de voto electrónico sea efectivo debe contar con la mayor capacidad de recursos técnicos que lo hagan fiables y para ello las herramientas existen. Podemos disponernos a diagramar presentándolo en distintas estructuras.

Por un lado, la Base de datos que debe contener distintos registros organizados según su dependencia en datos personales, censo electoral, votaciones y resultados. Así mismo, la base de datos, contiene un gestor de usuarios con programas que realicen tareas como registro, alta, baja y modificación de usuarios, control de contraseñas y tarjetas de voto, seguimiento, seguridad y autenticación e interacción con el correo electrónico. También un gestor de consultas que permita la creación de las mismas, validación de usuarios y votos, recuento de votos y presentación de resultados – estos pueden ser foros de debate, chats y comunicaciones en línea o por correo electrónico.

Las herramientas de las que disponemos para que la estructura del voto

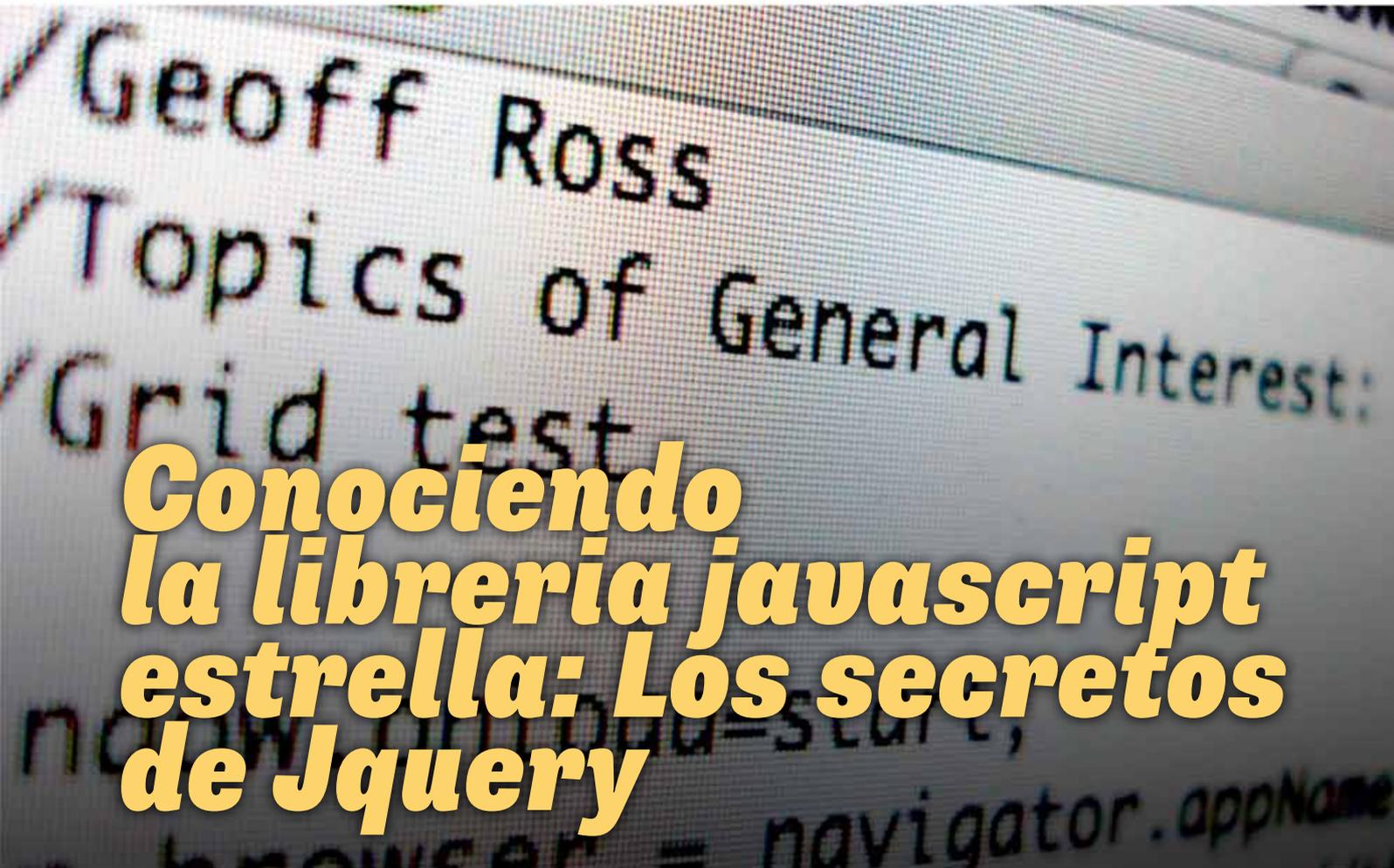
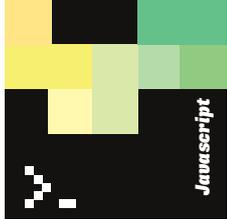
electrónico funcione serán la web y el lenguaje html. Es recomendable dejar de lado los plug-ins, Javascript y applets java. Por tanto, el servidor de base de datos deberá tener capacidad para la transacción de estos. Si buscamos una opción libre pues tenemos el PostGreSQL-7X, bajo licencia BSD. Como servidor web lo

recomendable es usar Apache, por ser libre y por la seguridad y estabilidad que ofrece. Agregaremos soporte para SSL y de scripts CGI ya que ese es el lenguaje recomendable para unir el sistema web con el servidor de base de datos. Para el desarrollo libre del entorno serán bienvenidos FreeBSD o GNU/Linux. **P**



*En la Argentina, en el año 2007, la empresa Altec implementó el voto electrónico en la ciudad rionegrina de Las Grutas. Dos de las urnas impidieron sufragar al 25% del padrón de mujeres y cuatro resultaron con cero votos. El municipio terminó por abolir el método.”*





# Conociendo la librería javascript estrella: Los secretos de JQuery

Por Gabriel Walter Gaitano Ornia

[gabriel.gaitanoornia@pixelscode.com](mailto:gabriel.gaitanoornia@pixelscode.com)  
[@cege74](https://twitter.com/cege74)



## **1-LOS COMIENZOS**

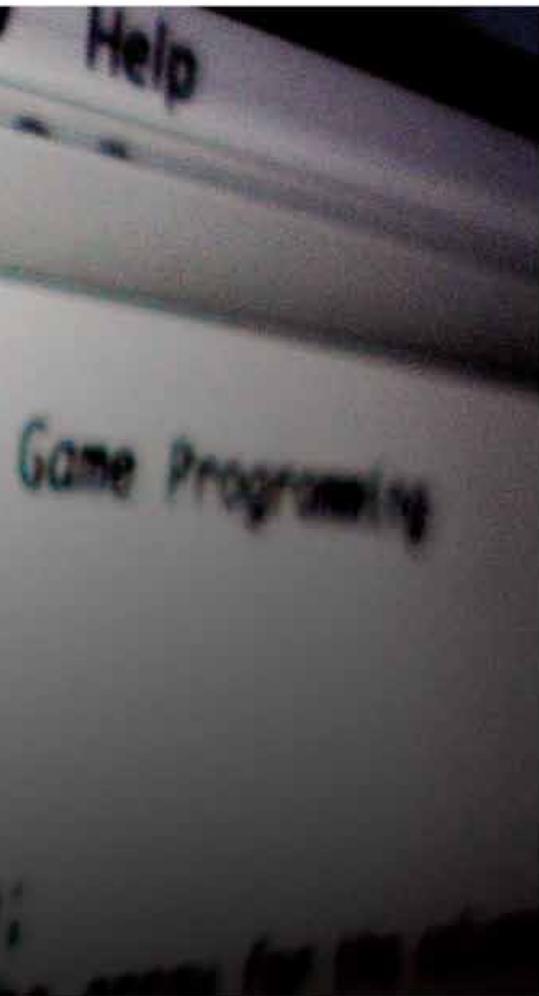
Las razones subyacentes al surgimiento de las librerías javascript son varias, pero esencialmente se vinculan a las guerras de los navegadores. Estas guerras, que se libraron y aún se libran en menor escala, se basaban en los distintos modos de implementar (o ignorar) los estándares web en las distintas empresas creadoras de navegadores. El ejemplo paradigmático fue la batalla entre el Internet Explorer de Microsoft y el Netscape Navigator.

Con ese motivo distintos programadores y grupos de programadores comenzaron a buscar modos de poder unificar y simplificar el desarrollo web. La creación de librerías fue una de las maneras que encontraron de escribir código web que pudiera verse de un modo estándar en todos los navegadores, o al menos que minimizara las incompatibilidades subyacentes. Así fueron surgiendo Mootools, Dojo, Script.aculo.us, Ext Js, Yahoo, y la que nos interesa particularmente, jQuery, entre muchas otras. Esta surgió a

principios del 2006 de la mano de John Resig, y muy pronto creció hasta ser aceptada y usada por un porcentaje muy alto de diseñadores y programadores en todo el mundo. Para dar una idea de su amplia difusión nos basta con comentar que Microsoft la adoptó en su interfaz propietaria de desarrollo Visual Studio, así como Nokia en su plataforma.

## **¿Y QUÉ ES JQUERY?**

Podemos definir a jQuery como una biblioteca escrita en el lenguaje



Javascript y pensada para escribir algunas operaciones típicas en la programación web en forma simple y compatible. Recordemos que Javascript es el lenguaje por excelencia de la web: íntimamente unido al HTML y a las hojas de estilo, permite controlarlos y manipularlos de un modo muy completo. jQuery está formado por una API (Application Program Interface: Interfaz de Programas de Aplicación) de funciones que permiten realizar programas que obvian las diferencias entre navegadores usando unas pocas líneas de código. A nivel de licencia es una biblioteca de código libre bajo la conocida GPL.

## 2-DESCUBRIENDO JQUERY

Para comenzar a descubrir esta formidable herramienta de desarrollo podemos realizar la descarga desde jquery.com, donde nos ofrecen una versión que ha sido minificada, es decir que no tiene comentarios ni espacios en blanco. Esto suele hacerse así para que la librería ocupe menos espacio a costa de legibilidad, pero de todos modos es posible bajarse la librería en formato extendido (legible), para cualquiera que desee explorar cómo está realizada y eventualmente agregarle funcionalidades nuevas. Es de destacar que al tratarse de un producto Open Source, los agregados y modificaciones deben ser puestos a disposición de la comunidad con la misma licencia GPL que tiene la librería original.

Una vez que la hayamos bajado y para poder utilizarla en alguna de nuestras páginas, es necesario cargarla como haríamos con cualquier librería de funciones Javascript. Si a modo de ejemplo la instalamos en una carpeta llamada librerías, la línea de carga sería:

```
<script src="librerías/jquery-latest.min.js"> </script>
```

A diferencia de otras librerías, en jQuery la mayoría de sus funcionalidades pueden ser accedidas usando la función \$. En el interior de la biblioteca se define un alias de la función jQuery a través de este símbolo. Esta función retorna un objeto de tipo jQuery sobre el cual es aplicada la función, proveyendo gran cantidad de posibilidades. Veamos, por ejemplo, el lanzamiento de código



*(...) es posible bajarse la librería en formato extendido (legible), para cualquiera que desee explorar cómo está realizada y eventualmente agregarle funcionalidades nuevas."*

cuando el documento está listo. Esta tarea es lo primero que la mayoría de los programadores Javascript hacen, agregando código a su programa de este modo:

```
window.onload = function()
{ alert("Iniciando carga del sistema"); }
```

Dentro de esto se suele ubicar el código que queremos ejecutar en el momento que la página se carga. Sin embargo, este enfoque tiene algunos inconvenientes, como veremos a continuación: principalmente nos encontramos con que el código javascript que escribamos allí no se ejecutará hasta que todas las imágenes sean a su vez cargadas, incluyendo los carteles de publicidad, los popularmente conocidos "banners". La razón que teníamos desde un principio para utilizar window.onload era que el documento HTML no se había cargado aun cuando intentamos correr nuestro código. ¿Qué solución nos aporta jQuery? Una simple instrucción que chequea el documento web y permanece

```
var Torrent = Class.create();
Torrent.prototype = {
  initialize: function(xml) {
    var xmlString = new XMLSerializer().serializeToString(xml);
    var torrentNode = xml.getElementsByTagName("torrent")[0];
    var statsNode = xml.getElementsByTagName("stats")[0];

    this.objectID = getText(torrentNode.getAttribute("id"));
    this.name = getText(torrentNode.getAttribute("name"));
    this.displayName = (this.name.length > 20 ? this.name.substr(0, 20) + "..." : this.name);
    this.hash = getText(torrentNode.getAttribute("hash"));
  }
};
```

“ Podemos definir a jQuery como una biblioteca escrita en el lenguaje Javascript y pensada para escribir algunas operaciones típicas en la programación web en forma simple y compatible.”

esperando hasta que esté listo para ser manipulado, el evento ready. Veámoslo en acción:

```
$(document).ready(function() {
  // el código que deseamos ejecutar va aquí
});
```

Ahora juguemos un poco, añadamos un gestor de evento para el click sobre un link en nuestra página de prueba:

```
$(document).ready(function() {
  $("a").click(function(event) {
    alert("¡Gracias por visitarnos!");
  });
});
```

Grabemos la página con estas modificaciones. Al clicar en el link, debería generarse una ventana emergente del navegador con nuestro mensaje, antes de ir a ejecutar el resto de la página, carga de imágenes incluida. Bien, veamos ahora algunas de las cosas que jQuery puede ayudarnos a hacer con nuestro sitio.

### 3-LOS PODERES DE LA LIBRERÍA

Hay muchas tareas que se nos presentan a diario en nuestro trabajo de diseño y programación. Vamos a intentar explorar algunas utilizando jQuery para empezar a notar su poder.

-Búsqueda de elementos en la página  
Para encontrar un elemento en el interior de una página, la forma habitual de hacerlo vía DOM sería, por ejemplo:

```
<p id="prueba">Probando</p>
<script>
var parrafo = document.getElementById( "prueba" );
...
</script>
```

Esto mismo utilizando jQuery lo hacemos de este modo:

```
<p id="prueba">Probando</p>
<script>
var para = $("#prueba");
...
</script>
```

Cuando a la función \$ le es pasado un string a modo de parámetro, este string se interpreta como un selector CSS normalmente del tipo .clase para identificar una clase, o .id para identificar un elemento que sea único. Cuando tenemos entonces el objeto jQuery, podemos realizar alguna tarea que necesitemos con él, veamos un ejemplo:

```
<script>
var para = $("#Prueba");
alert( para.text() ); // Muestra en una ventana "Prueba"
para.hide(); // Esconde el párrafo
</script>
```

Es posible también navegar a través del árbol XML del documento usando funciones como next(), prev(), parent(), children(), etc. Además, el objeto jQuery puede contener más de un elemento HTML, por ejemplo:

```
<div class="testc">Mi Div 1</div>
```

```
<div class="testc">Mi Div
2</div>
<div class="testc">Mi Div
3</div>
<script>
var divs = $(".testc");
// Esto podría ser también
leído como $(".div.testc")
divs.each( function(i) {
// i itera desde 0 hasta
divs.length-1
$(this).hide();
});
</script>
```

En este caso, para referirse a cada elemento dentro del objeto divs vamos a usar la función each, que recibe como parámetro otra función. En el ejemplo que vimos más arriba esta función es una función anónima, dado que no le damos ningún nombre. Dentro de la función, la variable de referencia this contiene el elemento, y si decidimos usar \$(this) obtenemos un objeto jQuery contenedor de este elemento.

-Modificar el marcado HTML de modo dinámico

Cuando finalmente completamos la selección del elemento que deseamos utilizar, o sea una vez que un elemento está elegido del modo que mostramos previamente, es posible cambiar el HTML albergado en el interior del elemento arbitrariamente. Lo que no debemos perder de vista al realizar esto es que lo que se especifique debe ser todo texto o bien HTML válido. Veamos a modo de ejemplos:

```
<div id="x">Bienvenido a
nuestro site</div>
<script>
$(".#id").html( "Adiós, te
```



```
extrañaremos" );
</script>
```

No existe la necesidad de especificar en este caso el elemento div; lo que estamos en definitiva reemplazando es solamente el HTML que está dentro del elemento seleccionado (el innerHTML, como lo solemos conocer). Además, si usamos la función sin parámetros html() esta retorna el contenido HTML de un elemento, attr("atributo") retorna un atributo, y text() retorna sólo el texto contenido en el elemento. Al retornar, también señalamos que lo deja disponible para su modificación.

También tenemos la posibilidad a nuestro alcance de agregar contenido antes o después del elemento seleccionado usando before() y after() respectivamente, o agregar contenido por dentro del elemento al principio o al final, usando prepend() y append(), o envolver al elemento usando wrap(), entre otras operaciones:

```
<div id="x">Hola</div>
```

```
<script>
$(".#x").before( "<p>A/</p>" ); // <p>A/<p><div
id="x">Hola</div>
$(".#x").after( "<p>A/</p>" ); // <div
id="x">Hola</div><p>A/</p>
```

```
$(".#x").prepend(
"<p>A/<p>" ); // <div
id="x"><p>A/<p>Hola</div>
$(".#x").append( "<p>A/</p>" ); // <div
id="x">Hola<p>A/</p></div>
```

```
$(".#x").wrap( "<div></div>" ); // <div><div
id="x">Hola</div></div>
</script>
```

-Tocar los estilos dinámicamente de la página

Nada nos impide modificar completamente el CSS de un elemento, agregando algunas propiedades o sobrescribiendo otras, por ejemplo:

```
<div id="x">Amarillo</div>
<script>
$("#x").css( { color:
"blue" } );
$("#x").css( { color:
"blue", fontSize: "120%",
backgroundColor: "#ffff00"
} );
</script>
```

Como ya vimos anteriormente, existen las funciones show() y hide() que, básicamente, actúan sobre el CSS de un elemento para mostrarlo u ocultarlo. La función toggle() las complementa puesto que muestra un elemento si este está oculto, y lo oculta si está visible.

También es posible usar una animación sobre estas funciones pasando un parámetro con valor "slow", "normal" o "fast", por ejemplo:

```
<div id="x"
style="display: none;">
Amarillo </div>
<script>$("#x").
show("slow");</script>
```

Otros efectos de animación incluyen la muy atractiva fadeOut(), que desvanece un elemento haciéndolo transparente, y fadeIn() que lo hace reaparecer a continuación. Una función más general para crear una animación a través de estas funciones es animate() que puede recibir dos parámetros, uno conteniendo un objeto con propiedades CSS para el final de la animación, y el segundo conteniendo la cantidad de milisegundos que debe durar la animación. Este ejemplo agranda un texto y lo desplaza hacia la derecha, demorán-



*No existe la necesidad de especificar en este caso el elemento div; lo que estamos en definitiva reemplazando es solamente el HTML que está dentro del elemento seleccionado (el innerHTML, como lo solemos conocer)."*

dose 3 segundos en el proceso:

```
<div id="x">Texto</div>
<script>
$("#x").animate( {
color: "#0000ff",
backgroundColor:
"#00ff00",
fontSize: "3em",
paddingLeft: "200px"
}, 3000);
</script>
```

- Agregar eventos a la página Agregar un evento a un elemento es muy fácil y lo mejor es que funciona en todos los navegadores de la misma manera. Las funciones para agregar eventos incluyen click(), dblclick(), mouseover(), mouseout(), etc. Viendo un ejemplo:

```
<p id="x">Prueba</p>
<script>
$("#x").click(
function(ev) {
alert( "Evento click: " +
ev.getSource() );
}
);
</script>
```

Este ejemplo, un poco más complejo, permite editar el contenido de un elemento HTML:

```
<code>
<p id="x">Cambiamelo</p>
<input id="z"
value="Cambiamelo"/>
<script>
$("#z").keyup(
function(ev) {
var newtext =
$(this).attr("value");
$("#x").text(
newtext );
});
</script>
</code>
```

#### 4- JQUERY Y AJAX

En jQuery se incorporan algunas funciones para hacer la programación en Ajax más simple. Todos sabemos la revolución que supuso el uso de esta tecnología en la usabilidad y capacidades de la Web, sin embargo en ocasiones su uso sigue siendo complejo. A través de jQuery se busca, en forma consecuente con su filosofía, conseguir tornar mucho más fácil su uso. Enviar datos a un servidor es muy fácil, hay dos funciones \$.get() y \$.post() que corresponden a los métodos HTTP GET (pasar parámetros en la URL) y POST (enviar parámetros en el cuerpo del request HTTP, útil cuando hay más datos). Ambas funciones reciben la URL donde está el servicio que se quiere llamar, opcionalmente los datos que se deben entregar al servicio y opcionalmente, también, una función que se debe eje-

cutar si la llamada en Ajax tiene éxito. En este ejemplo, asumimos que en el servidor hay un script server.php que contiene código para recibir los datos que estamos enviando, y que imprime a la salida un mensaje al recibirlos:

```
$.get( "server.php",  
  { nombre: "Juan Ignacio", puntos: "9" },  
  function(msg) {  
    alert( "Se grabaron los datos: " + msg );  
  }  
);
```

Pueden recibirse datos en XML también, e incluso usarse la misma función de selección de \$(" ") para seleccionar datos dentro del XML recibido, usando un segundo parámetro:

```
// El XML en 'datos.xml'  
contiene:  
// <?xml version="1.0">  
// <datos>  
//   <nombre>Juani</nombre><puntos>10</puntos>  
// </datos>
```

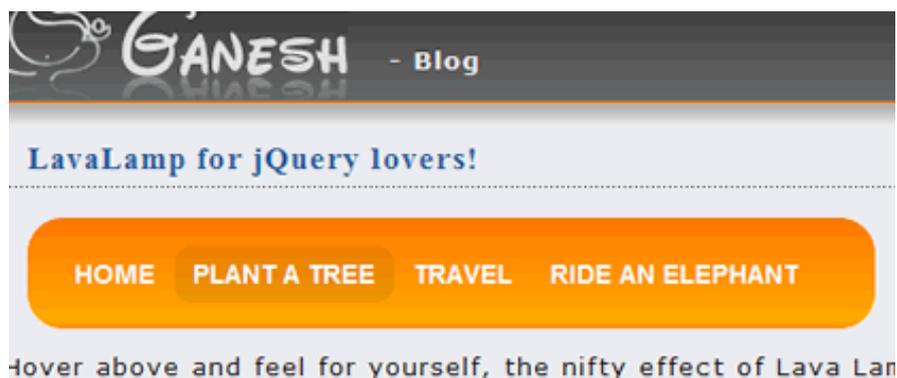
```
$.get( 'datos.xml',  
  {},  
  function(xml) {  
    alert( $("nombre",xml).text() ); // Imprime "Juani"  
  });
```

### 5-ALGUNOS EJEMPLOS DE JQUERY EN ACCIÓN

Para terminar esta incursión en esta potente tecnología, podemos mostrar algunos ejemplos del uso de jQuery. **P**

#### 1) Menú de lava

<http://www.gmarwaha.com/blog/2007/08/23/lavalamp-for-jquery-lovers/>



#### 2) Imágenes cambiantes

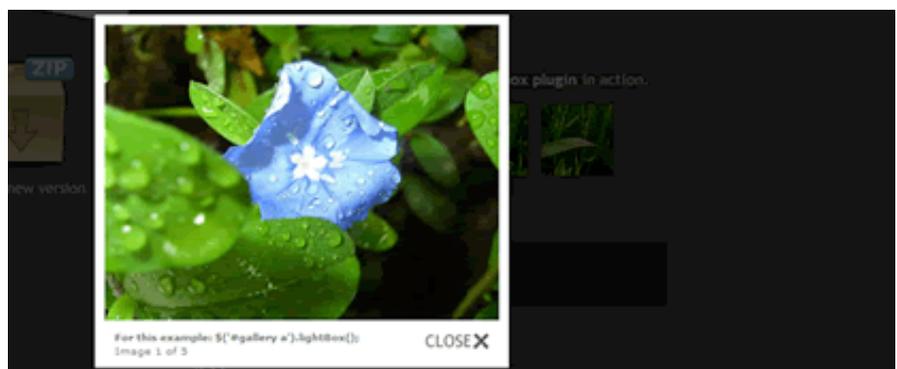
<http://medienfreunde.com/lab/innerfade/>

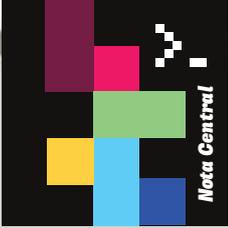
A list with images and links



#### 3) Imágenes resaltadas

<http://leandrovieira.com/projects/jquery/lightbox/>





# Desarrollo de juegos con HTML5

Por Matías Iacono

[matias.iacono@pixelscode.com](mailto:matias.iacono@pixelscode.com)

**E**l desarrollo de video juegos ha sido, desde los principios de la computación, algo que sólo ha sido trabajado por cierto grupo de desarrolladores que se separaban de la computación más tradicional. Posiblemente por este motivo, fue que siempre este tipo de desarrollos estuvo envuelto en un halo de misticismo o simplemente debido a que la información disponible para desarrollar juegos no abundaba tanto como otras. En todo caso, con el correr del tiempo, la información ha ido acrecentándose y las empresas encargadas de la creación de herramientas aplicables al desarrollo de juegos también han proporcionado las vías para que podamos, en la actualidad, incursionar sobre esta área de forma más simple y rápida.

De cualquier manera, la evolución natural de los lenguajes y tecnologías han hecho que dejemos de lado algunas plataformas objetivo de desarrollo, nos mudemos a otras, cambiemos los lenguajes, o esperemos ansiosos actualizaciones de APIs o IDEs.

Justamente esta tendencia evolutiva, así como el mudarnos de plataformas objetivo como de lenguajes de programación es algo que en el desarrollo de juegos volvemos a vivir, en este caso, de la mano

de HTML5 como nuevo estándar HTML y JavaScript, este último, como un nuevo viejo lenguaje de programación para el soporte y la creación de contenido dinámico.

### HTML5

Antes de que veamos algunos ejemplos de cómo podemos crear nuestros propios video juegos con HTML5 es necesario que veamos a qué hacemos referencia cuando hablamos de este estándar.

Lo primero que debemos de destacar es que HTML5, hasta la fecha, no es un estándar cerrado y aprobado. Esto quiere decir que en el proceso de aceptación, implementación y estandarización final, el organismo encargado de dar el visto bueno sobre el mismo aún no lo ha hecho, por lo tanto, es posible que todo desarrollo que encaremos hoy pueda no necesariamente comportarse como esperábamos una vez el estándar esté cerrado y sea público. De cualquier manera, el mismo se encuentra en etapas muy maduras, por lo que se asume que los cambios, de haberlos, no deberían impactar significativamente en lo que ya se haya construido.

Podemos ver mucho más sobre el estándar HTML5 y todos sus cambios desde la página oficial de la W3C (<http://dev.w3.org/html5/>



*Lo primero que necesitaremos*

*hacer será crear una página Web e incluir un tag <canvas> dentro de esta que nos servirá de lienzo.”*

[spec/Overview.html](#)) y en el número anterior de Pixels&Code.

A pesar de que este estándar ya casi esté terminado, no todos los navegadores Web lo soportan, incluso muchos navegadores (de diferentes empresas) han ido creando nuevas versiones de forma rápida para poder soportar HTML5 y no dejar relegado al usuario a las posibilidades del mismo. Esto quiere decir que si optáramos por este estándar para desarrollar contenido, nos podremos encontrar que nuestros usuarios aún no cuenten con el navegador adecuado para poder visualizarlo.

Es posible, si nunca hemos desarrollado o escuchado sobre este modelo de desarrollo, que nos estemos preguntado: ¿Qué es HTML5? Esta pregunta puede surgir en especial por la mezcla de las siglas HTML y el número, como si se tratase de alguna tecnología nueva propiedad de alguna empresa similar a Flash o Silverlight. Lo cierto es que HTML5 no es más que HTML (Hypertext Markup

## >NOTA CENTRAL-

Language), esto es, el código de maquetación que se utiliza en el diseño y creación de páginas Webs tradicionales y su versión. Para tener una idea más interesante, es posible que hasta la fecha hubiéramos estado utilizando HTML4 sin saber de ello. De cualquier manera, HTML5 intenta incorporar al HTML componentes o etiquetas que se consideran en la actualidad básicas y necesarias para cualquier sitio Web. Si nos remitimos un poco a la historia del desarrollo Web podremos encontrar, en sus principios, etiquetas permitidas dentro del HTML como la conocida <blink> o <marquee>, haciendo que un texto parpadeara para el caso de la primera o se desplazara como una marquesina para la segunda. Estas etiquetas resultaban útiles en tiempos donde los navegadores eran primitivos y el soporte dinámico, así como de componentes adicionales, no existían. Pero, requería que los

navegadores implementaran esta funcionalidad para dar soporte a las mismas. Con el tiempo, y con avance tecnológico, hemos podido apreciar que el uso de video o material dinámico en los sitios resulta natural. Esta naturalidad es la que ha llevado y lleva a que el estándar evolucione para incorporar estas características dentro del mismo código HTML. Una de las tantas ventajas que podríamos enumerar es la que no dependeríamos de componentes de terceros para determinadas características por lo que, por ejemplo, la reproducción de video desde una página Web estaría asegurada por implementación del navegador y no debido a que un usuario requiera tener la última actualización de un componente de terceros como es el caso de Adobe Flash.

A continuación podemos encontrar algunos de los tags más interesantes de HTML5.

TAG	DESCRIPCIÓN
<canvas>	Representa un lienzo para dibujar. Desde JavaScript será uno de los principales tags a utilizar en el desarrollo de juegos.
<audio>	Este tag permite la ejecución de sonidos y música en formatos tales como MP3, Ogg y Wav.
<video>	Reproduce videos mediante streaming en formatos Ogg, MPEG4 y WebM

Podemos encontrar una lista completa en la dirección [http://www.w3schools.com/html5/html5\\_new\\_elements.asp](http://www.w3schools.com/html5/html5_new_elements.asp)



*(...) las teclas de dirección del teclado modificarán la posición en X e Y de la imagen dando la idea de control sobre la misma.”*

HTML5 presenta muchos otros nuevos tags o algunos revisados e importados desde HTML4, en todo caso, los listados en la tabla son aquellos a los que como desarrolladores de juegos daremos mayor uso.

### JAVASCRIPT

Conociendo ya qué es HTML5, debemos hablar un poco sobre el lenguaje de programación que dará vida a nuestros juegos. En este caso, y recordando lo que decíamos al inicio, muchos lenguajes que en algún punto no se los consideró muy importantes, vuelven a reciclarse para ser utilizados con un nuevo fin. Si bien JavaScript siempre ha estado asociado a la Web, no siempre fue bien visto o tan usado como se podría esperar. A pesar de ello, desde la creciente implementación y uso de A.J.A.X. (Asynchronous JavaScript and XML) en los sitios Web, este lenguaje se ha visto reivindicado y siendo que es el único plenamente soportado por los distintos navegadores y con contacto directo para la manipulación de las etiquetas HTML, ha pasado a ser el predilecto

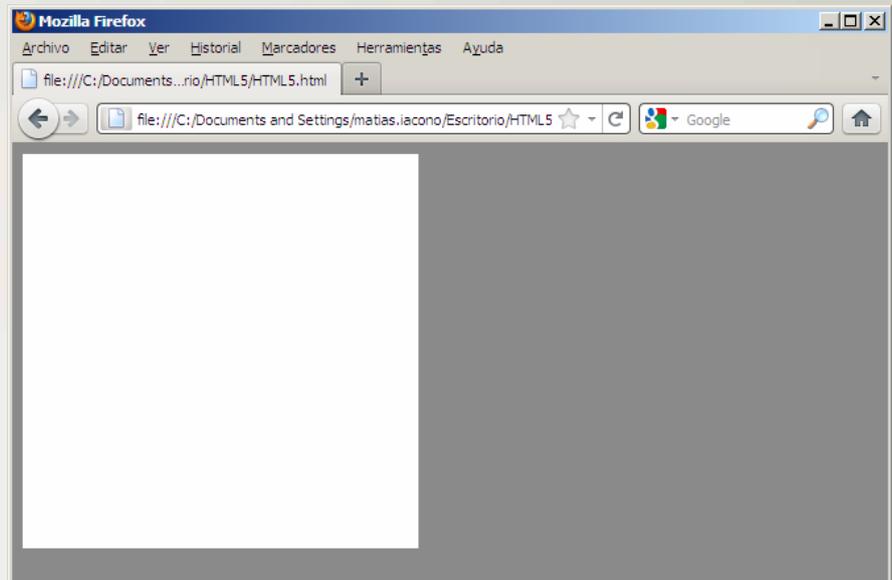
para la unión con HTML5. Sólo adicionaremos, para aquellos que no sepan nada de JavaScript, que este lenguaje presenta una estructura similar a los lenguajes basados en C y C++ aunque con la particularidad de ser considerado dinámico. Veremos en este artículo cómo implementar algunas líneas mínimas para establecer las pautas iniciales para el desarrollo de juegos.

### PRIMEROS PASOS

En un artículo anterior, sobre XNA, habíamos visto rápidamente que uno de los componentes necesarios para el desarrollo de juegos era la de un lienzo donde dibujar nuestros gráficos. Otro, un bucle que se repetía continuamente para dibujar, cuadro a cuadro, las animaciones del juego así como actualizar la posición de cada uno de los objetos del juego. Si bien con JavaScript y HTML5 contamos con todo esto, tendremos que hacer un trabajo más minucioso para la implementación de toda la lógica de ejecución de nuestro juego. Esto se debe a que no contamos con un Framework (Marco de trabajo) nativo para el desarrollo de juegos bajo esta tecnología.

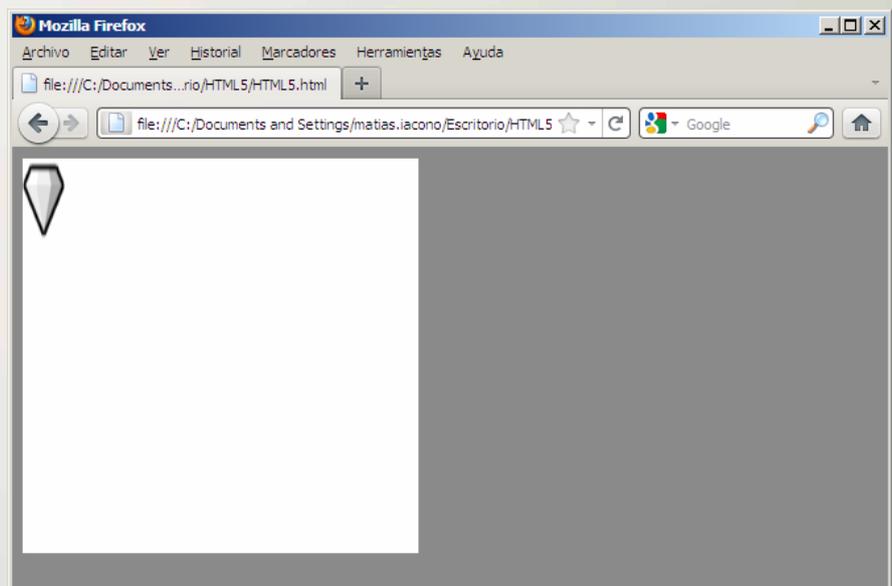
Lo primero que necesitaremos hacer será crear una página Web e incluir un tag `<canvas>` dentro de esta que nos servirá de lienzo. Podemos usar cualquier editor de texto para la creación de esta página.

```
<html>
<head></head>
<body style="background-color:gray">
  <canvas id="lienzo"
  style="width:300px;
```



```
height:300px;
background-
color:white"></canvas>
</body>
</html>
```

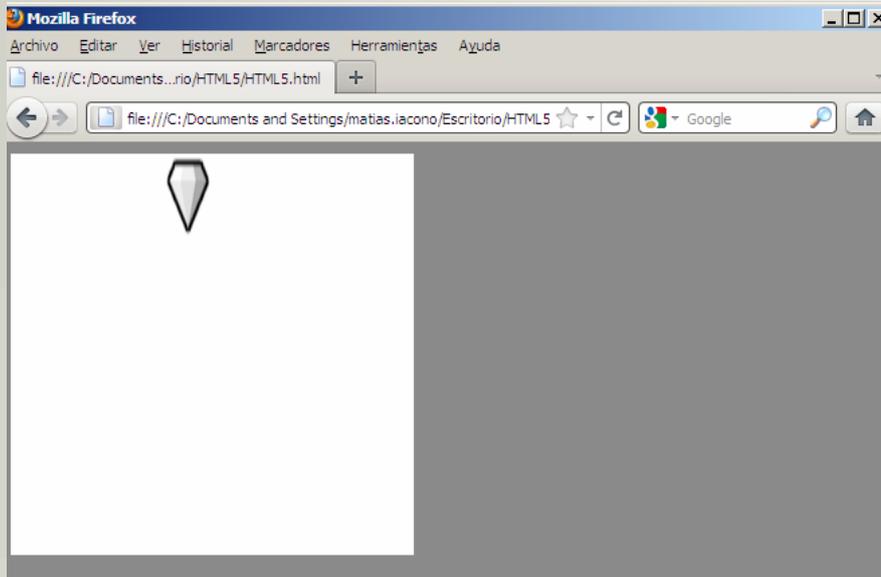
*El lienzo listo para ser utilizado en nuestros juegos.*



El código anterior nos proporciona una visión aproximada de cómo se verá nuestra página y el lienzo de dibujo de nuestro juego.

*Nuestra primera prueba de dibujo sobre el objeto Canvas.*

## >NOTA CENTRAL-



Al dejar trabajar a nuestro código podemos ver cómo la posición de la imagen es modificada y nos da la sensación de que se está moviendo.

El siguiente paso será la creación de código JavaScript para la manipulación de este lienzo de dibujo. Para esto tendremos que capturar el lienzo desde el código, especificar el formato de dibujado y, por último, enviar alguna imagen al mismo.

```
<script
language="javascript">
  var can-
vas = document.
getElementById("lienzo");
  var context = can-
vas.getContext("2d");
  ...
  ...
</script>
```

El código JavaScript anterior es clave en la creación de este pequeño motor para nuestros juegos. La primera línea de código captura el objeto <canvas> que habíamos colocado anteriormente en la página HTML mediante el atributo ID. Esto nos dará acceso a las funcionalidades de este objeto como,

por ejemplo, la que vemos en la segunda línea. En esta segunda línea obtenemos un objeto de dibujo sobre el lienzo previamente capturado. Aquí el tipo de objeto será del tipo 2D (bidimensional), esto es, con coordenadas X e Y para el dibujado de diferentes figuras. El siguiente paso es contar con las imágenes de nuestro juego, cargarlas y dibujarlas cuando sea propicio.

```
var imagen = new Image;
imagen.src = "Gem.png";
context.drawImage(imagen,
0, 0, imagen.width, ima-
gen.height);
```

Estas líneas de código podríamos colocarlas a continuación de las anteriores. Lo que harán será, inicialmente, leer desde la dirección especificada una imagen. Para tal fin podemos utilizar cualquier formato soportado en la Web, en nuestro ejemplo usamos una imagen en formato PNG (Portable Network Graphic), debido a que

este ya incluye transparencias, lo que hará que no se vea un recuadro alrededor de la imagen sino sólo su contenido. La última línea es posiblemente la que más utilizaremos cuando estemos creando nuestros juegos. Esta dibuja directamente la imagen sobre el objeto de dibujo, mismo que vuelca estos cambios al objeto <canvas> para mostrarse en pantalla.

Por supuesto, esto no es un juego en sí mismo, pero nos da una idea para el siguiente paso. Como en todo juego, un elemento fundamental es el de crear un bucle constante, que no se detenga y que por cada ciclo todos los objetos del juego sean actualizados y redibujados. En JavaScript podríamos crear esto de forma simple con el siguiente código.

```
var x = 0;
setInterval(Dibujar, 1 /
30);
function Dibujar() {
  context.fillStyle =
```

```

"rgb(255,255,255)";
    context.fillRect(0,
0, canvas.width, canvas.
height);
    context.
drawImage(imagen, x, 0,
imagen.width, imagen.
height);
    x++;
}

```

La función setInterval de JavaScript nos permite llamar a una función específica cada cierto tiempo definido. En este caso, llamaremos a la función Dibujar(), la que se encargará de redibujar el lienzo y modificar la posición de nuestro gráfico en base a la coordenada X.

El siguiente y último paso para tener un esquema inicial de un juego es el de capturar algún periférico, como el teclado o el ratón, y dejar que el usuario interactúe con los objetos del juego.

```

document.onkeydown =
function(event) {
    if (event.keyCode ==
37)
        x--;
    if (event.keyCode ==
39)
        x++;
    if (event.keyCode ==
38)
        y--;
    if (event.keyCode ==
40)
        y++;
}
function Dibujar() {
    ...
    context.

```

```

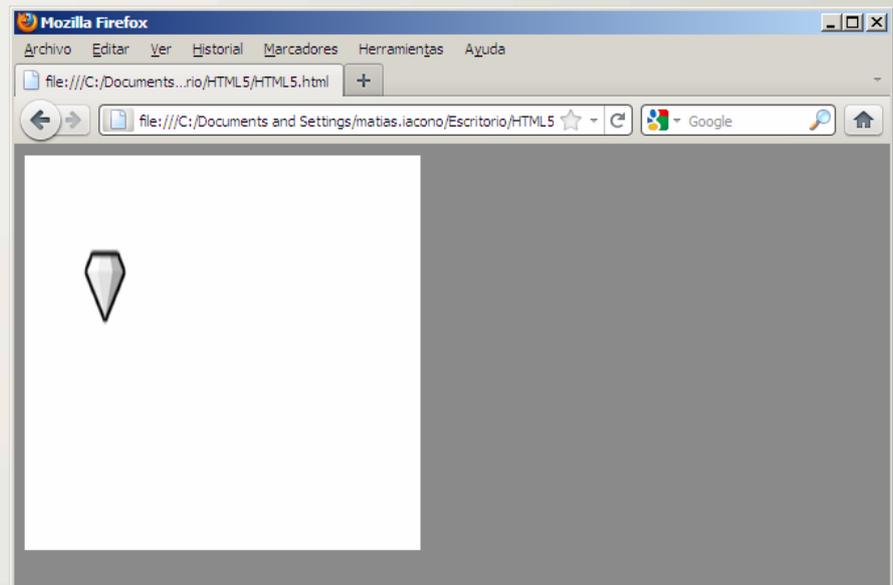
drawImage(imagen, x, y,
imagen.width, imagen.
height);
}

```

Agregando esta nueva función y modificando un poco la función dibujar es posible capturar, de forma rudimentaria, las teclas que el usuario presiona y modificar las coordenadas de nuestro objeto en base a esto. En el ejemplo, las teclas de dirección del teclado modificarán la posición en X e Y de la imagen dando la idea de control sobre la misma.

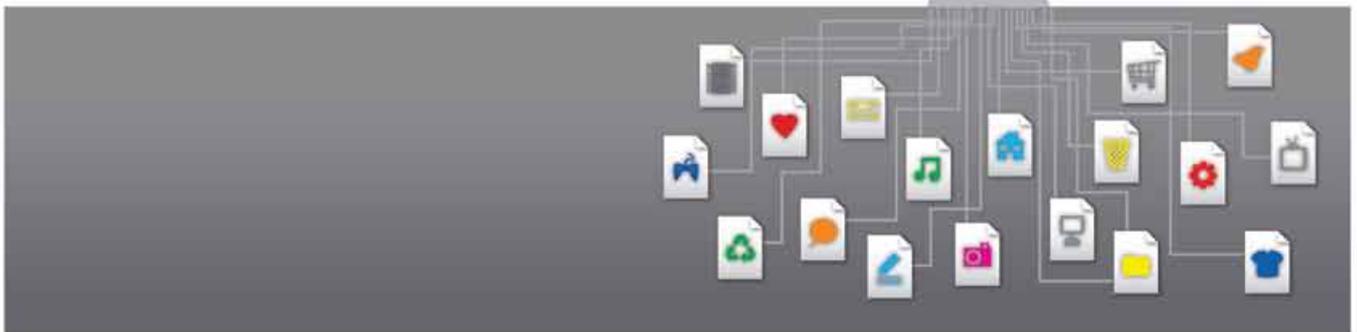
Indudablemente, lo que hemos realizado hasta ahora no podríamos

considerarlo un juego completo, pero nos puede dar una idea para intentar iniciarnos en este mundo. Es claro que además deberemos agregar sonidos, más objetos y un mejor control de la interacción con el teclado o el ratón. De cualquier manera, en el mercado existe una gran cantidad de Frameworks para el desarrollo de juegos con HTML5, algunos gratuitos y otros no tanto, por lo que podremos optar por crearnos uno propio o seleccionar uno de estos. En todo caso, en futuras entregas iremos adicionando mayor funcionalidad a este boceto inicial y así tener una mejor idea de la estructura del desarrollo de juegos con JavaScript y HTML5. **D**



*Al presionar algunas de las teclas, el objeto gráfico del juego se desplazará dando la sensación de movimiento clásico en un video juego.*

# Algoritmos: Binary Tree Sort



Por Juan Gutmann .....  
[juan.gutmann@pixelscode.com](mailto:juan.gutmann@pixelscode.com)

*En esta oportunidad vamos a analizar un algoritmo de implementación sencilla, pero de una dificultad conceptual más importante que algunos de los expuestos hasta el momento. Consiste en armar una estructura de datos llamada “árbol binario”, a partir de la lista que se desea ordenar. El mismo sistema con el que se irá confeccionando esta estructura genera el ordenamiento, por lo que bastará con posteriormente recorrer el árbol para obtener una versión de la lista ya ordenada. Por ello, es extremadamente eficiente, incluso para listas que contienen una cantidad elevada de elementos.*

## **EL ÁRBOL BINARIO**

Con este nombre llamamos a una estructura que contiene una cierta cantidad de elementos, conocidos como “Nodos”. Cada nodo a su vez puede estar enlazado a otros dos: un primer nodo “hijo”, que podemos llamar “izquierdo”, y un segundo nodo hijo, el “derecho”. Para generar el árbol, se toma un elemento cualquiera de la lista de valores que se quiere almacenar en esta estructura (por lo general se suele elegir el primer valor). Este elemento será la “raíz”. A partir de

allí, se sigue el siguiente criterio: tomamos otro valor de la lista, y nos fijamos si es menor que el valor del nodo actual (en la primera pasada, el nodo actual será la “raíz” del árbol). En caso afirmativo, almacenamos este valor en el nodo

hijo “izquierdo”. De lo contrario, lo hacemos en el nodo hijo “derecho”. Para los valores subsiguientes, en el caso de que sea menor que la raíz, evaluamos si ya existe un nodo hijo izquierdo. Si esto es verdad, este nodo hijo pasa a ser el nodo actual, y se repite el proceso. Nos manejamos de idéntica forma si el valor es mayor que el contenido en el nodo actual, pero con el nodo hijo “derecho”. Repetimos este método para los restantes valores de la lista a ordenar, hasta haber almacenado todos en el árbol. Como ya habrán podido adivinar, el algoritmo para generar esta estructura es de tipo recursivo, un concepto que ya hemos aplicado más de una vez en esta sección. Cuando concluye este proceso, si contemplamos la estructura obtenida invertida en forma vertical, comprendemos la razón de su nombre: del primer nodo (la raíz) derivan todos los demás, como “hojas” que cuelgan de “ramas”: de allí el empleo de la palabra “árbol”. Como de cada nodo pueden colgar solamente otros dos nodos, es un árbol de tipo “binario”.

### **EL BINARY TREE SORT**

Para obtener una lista ordenada en base a un árbol binario, aplicamos un algoritmo conceptualmente muy similar al utilizado para generarlo. Su funcionamiento también es recursivo. Este algoritmo recibe un nodo como parámetro. Si el nodo no tiene hijos, quiere decir que hemos llegado a una de las “puntas” del árbol, por lo cual se retorna una lista conteniendo un único elemento: el valor de dicho nodo. Si por el contrario, tiene hijos, retornamos una lista constituida de la siguiente manera: la concatenación de la lista devuelta por este mismo algoritmo, aplicada sobre el nodo hijo izquierdo (si existe), el valor del nodo propiamente dicho, y la lista devuelta por el algoritmo aplicado sobre el nodo hijo derecho (nuevamente, si lo hubiera). La repetición recursiva de este código terminará devolviendo la lista completa, ya ordenada.

Veamos en primer lugar el pseudocódigo para armar el árbol:

Dada una lista L:

Insertar en el árbol un Nodo con el valor del primer elemento de L (raíz)

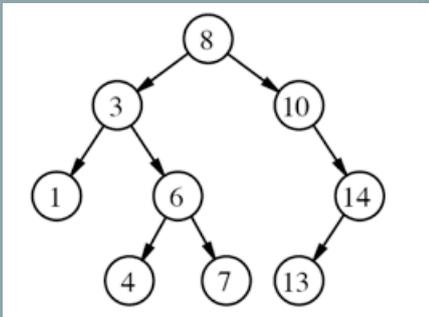
```

para todos los elementos restantes de L
  nodo actual = raíz
  evaluar:
  si valor del elemento < valor de nodo actual
    si (no existe nodo hijo izquierdo del nodo actual)
      crear nodo hijo izquierdo con valor del elemento
    si no
      nodo actual = nodo hijo izquierdo
      repetir evaluar
  fin si
si no
  si (no existe nodo hijo derecho del nodo actual)
    crear nodo hijo derecho con valor del elemento
  si no
    nodo actual = nodo hijo derecho
    repetir evaluar
  fin si
fin si
fin para

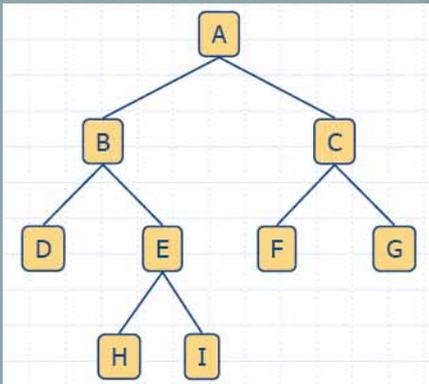
```

Ahora, el pseudocódigo correspondiente al algoritmo que recorre el árbol, devolviendo una lista con los elementos ordenados: Dada una función arbol\_a\_lista, que recibe como argumento un Nodo N y devuelve una lista:

## >ALGORITMOS-



*binary\_tree.jpg: Un clásico ejemplo de árbol binario.*



*binary\_tree\_alfa.jpg: Otra versión de árbol binario, que en este caso almacena caracteres alfabéticos.*

```
J = lista vacía
```

```
si (N no tiene hijos)
```

```
    agregar a J un elemento con el valor de N
```

```
si no
```

```
    si N tiene hijo izquierdo
```

```
        J = J + arbol_a_lista(hijo izquierdo de N)
```

```
    fin si
```

```
    agregar a J un elemento con el valor de N
```

```
    si N tiene hijo derecho
```

```
        J = J + arbol_a_lista(hijo derecho de N)
```

```
    fin si
```

```
fin si
```

```
devolver J
```

Una vez armado el árbol, bastará con llamar a esta función, pasándole como parámetro el nodo raíz, para que realice el trabajo en forma recursiva, devolviendo como resultado una lista conteniendo los elementos ordenados.

### **IMPLEMENTACIÓN EN PYTHON**

En esta oportunidad, vamos a agregar a la versión en Python una doble salida en pantalla, para poder comprender mejor la forma de trabajo del Binary Tree Sort. Por un lado, vamos a mostrar en cada llamado a insertar() la constitución de cada nodo: el valor del mismo nodo, y los de sus hijos izquierdo y derecho, si es que existen. Pero también vamos a escribir un pequeño método recursivo al que llamaremos una vez completado el armado del árbol. El mismo presentará una versión gráfica del árbol binario, rotado 90 grados a la izquierda para facilitar la programación del mismo. Para cada nodo, se mostrarán debajo de él su hijo izquierdo, y arriba el hijo derecho. Vale la pena destacar que en esta implementación hemos recurrido a un concepto tomado del paradigma de Programación Orientada a Objetos (OOP, siglas en inglés de "Object Oriented Programming"), definiendo una clase "Nodo" con tres atributos: valor, que contendrá el valor del nodo propiamente dicho, "izq" que alojará una referencia (los programadores de C y C++ pueden interpretar esta palabra como sinónimo de "puntero") al nodo hijo izquierdo, y "der" que hará lo mismo con el nodo hijo derecho.

```
class Nodo:
    def __init__(self, valor):
        self.izq = None
        self.der = None
        self.valor = valor
```

```
def insertar(N, valor):
    if valor < N.valor:
```

```

        if N.izq == None:
            N.izq = Nodo(valor)
        else:
            insertar(N.izq, valor)
    else:
        if N.der == None:
            N.der = Nodo(valor)
        else:
            insertar(N.der, valor)

    strPrint = "Nodo: " + str(N.valor) + " "

    if N.izq <> None:
        strPrint += "Hijo Izquierdo: " + str(N.izq.
valor) + " "

    if N.der <> None:
        strPrint += "Hijo Derecho: " + str(N.der.va-
lor) + " "

    print strPrint

def arbol_a_lista(N):
    J = []

    if N.izq == None and N.der == None:
        J.append(N.valor)
    else:
        if N.izq <> None:
            J = J + arbol_a_lista(N.izq)

        J.append(N.valor)

        if N.der <> None:
            J = J + arbol_a_lista(N.der)

    return J

def treeprint(N, nivel):
    ret = ""

    if N.der <> None:
        ret += treeprint(N.der, nivel + 1)

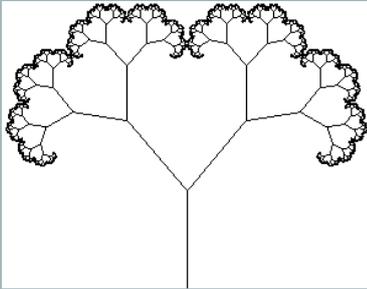
    ret += "\n" + (nivel * " ") + str(N.valor)
    if N.izq <> None:
        ret += treeprint(N.izq, nivel + 1)

    return ret

```

“ (...) tomamos otro valor de la lista, y nos fijamos si es menor que el valor del nodo actual (en la primera pasada, el nodo actual será la “raíz” del árbol). En caso afirmativo, almacenamos este valor en el nodo hijo “izquierdo”. De lo contrario, lo hacemos en el nodo hijo “derecho”.”

## >ALGORITMOS-



*fractal.gif: Prácticamente no existen los “geeks” informáticos que no amen los fractales. Lo interesante del que vemos aquí es que ha sido generado en base a una estructura de árbol binario.*

|||||tree-c.png. Esta es

```
#include<stdlib.h>
#include<stdio.h>

struct tree_n1 {
    int val;
    struct tree_n1 * right, * left;
};

typedef struct tree_n1 node;

void insert(node ** tree, node * item) {
    if(!(*tree)) {
        *tree = item;
        return;
    }
    if(item->val < (*tree)->val)
        insert(&(*tree)->left, item);
    else if(item->val > (*tree)->val)
        insert(&(*tree)->right, item);
}

void printout(node * tree) {
    if(tree->left) printout(tree->left);
    printf("%d\n", tree->val);
    if(tree->right) printout(tree->right);
}

void main() {
    node * curr, * root;
    int i;
    root = NULL;

    for(i=0; i<=10; i++) {
        curr = (node *)malloc(sizeof(node));
        curr->left = curr->right = NULL;
        curr->val = rand();
        insert(&root, curr);
    }

    printout(root);
}
```

*una implementación del algoritmo que genera el binary tree escrita en lenguaje C.*

|||||

```
def treesort(L):
    raiz = Nodo(L[0])

    for I in range(1, len(L)):
        insertar(raiz, L[I])

    print treeprint(raiz, 0)
    print

    return arbol_a_lista(raiz)
```

```
Z = [5, 4, 8, 2, 1, 3, 9, 7, 10, 6, 10]
print treesort(Z)
```

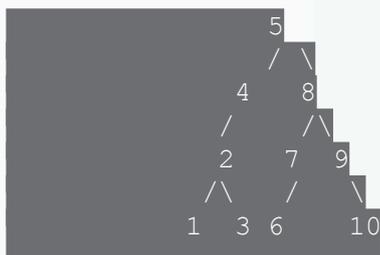
La salida por pantalla de esta versión en Python es la siguiente:

```
Nodo: 5 Hijo Izquierdo: 4
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
Nodo: 4 Hijo Izquierdo: 2
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
Nodo: 2 Hijo Izquierdo: 1
Nodo: 4 Hijo Izquierdo: 2
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
Nodo: 2 Hijo Izquierdo: 1 Hijo Derecho: 3
Nodo: 4 Hijo Izquierdo: 2
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
Nodo: 8 Hijo Derecho: 9
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
Nodo: 8 Hijo Izquierdo: 7 Hijo Derecho: 9
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
Nodo: 9 Hijo Derecho: 10
Nodo: 8 Hijo Izquierdo: 7 Hijo Derecho: 9
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
Nodo: 7 Hijo Izquierdo: 6
Nodo: 8 Hijo Izquierdo: 7 Hijo Derecho: 9
Nodo: 5 Hijo Izquierdo: 4 Hijo Derecho: 8
```

```
10
9
8
7
6
5
4
3
2
1
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Una representación vertical (sin rotar) del árbol, más trabajosa de programar, pero mucho más fácil de visualizar rápidamente, con los valores de la lista de prueba arrojaría lo siguiente:



### ASPECTOS TEÓRICOS

Para medir la eficiencia del algoritmo, deben considerarse tanto los recursos empleados para generar el árbol, como para recorrerlo posteriormente. Generar un árbol binario de  $N$  ítems tiene una complejidad de  $O(N \log(N))$ , lo que torna al binary sort tree en un método bastante más eficiente que otros de los que hemos expuesto en las entregas anteriores. En cuanto a la complejidad del algoritmo que recorre el árbol y devuelve la lista ya ordenada, la misma es de  $O(N)$ . Respecto de las optimizaciones posibles, existen algunas verdaderamente sofisticadas, pero una de las más simples que pueden aplicarse es la misma que la ya propuesta para la elección del pivot cuando analizamos el algoritmo Quick Sort: en lugar de tomar el primer elemento de la lista como nodo raíz del árbol, evaluamos tres elementos (por ejemplo, el primero, el segundo y el último), escogiendo como raíz al valor intermedio. De esta forma se logra una reducción en la cantidad de niveles (ramas) que tendrá el árbol una vez completado, y por ende llevará menor tiempo tanto su generación como su posterior recorrida.

### CONCLUSIÓN

Lo interesante de lo expuesto en esta edición es que el árbol binario puede aplicarse a la solución de muchos problemas complejos en programación. Uno de los usos más comunes en los que se aplican estructuras similares es en la compresión de archivos, siendo vitales para ayudar a concretar una de las tareas más difíciles que puede emprender un programador. Sin ir más lejos, el célebre Sistema Huffman, creado por el entonces alumno del MIT David A. Huffman en 1951, se basa en este concepto. Se trata de un notable algoritmo de compresión, muy eficiente cuando se lo emplea con archivos de texto. De esta forma, pueden ver que se cumple la promesa que les hicimos cuando dimos juntos los primeros pasos en esta serie de notas: los algoritmos de ordenamiento no solamente nos sirven para lo que fueron concebidos, sino que nos dan las herramientas para mejorar nuestra capacidad como programadores en general. **P**



Generar un árbol binario

de  $N$  ítems tiene una complejidad de  $O(N \log(N))$ , lo que torna al binary sort tree en un método bastante más eficiente que otros de los que hemos expuesto en las entregas anteriores.”



# Guía para desarrollar aplicaciones para BlackBerry

*El mundo de los negocios tiene un gran número de usuarios de teléfonos inteligentes que se sirven de sus utilidades tanto para trabajar como para su vida cotidiana. Aquí conoceremos una sencilla forma de crear aplicaciones útiles para aquellos que combinan las obligaciones con el entretenimiento, y prefieren tener todo al alcance de sus manos.*

Por Roxana Miguel

roxana.miguel@pixelscode.com  
@roxanamiguel



**B**lackBerry es eso, una línea de smartphones desarrollados por la compañía Research In Motion (RIM) que integra entre sus servicios el correo electrónico, hecho que lo convirtió en el dispositivo más popular en el mundo de los negocios, donde estar conectados es más que útil. Sin embargo, por su acceso a la red los usuarios lo eligieron como un modo de comunicación instantánea sin necesidad de mantener comunicaciones telefónicas de voz.

Sin duda, desde su aparición sobre finales de los años noventa, mucha ha sido la competencia que se cruzó en su camino, y razón de ello es que muchos desarrolladores nos ocupemos de crear aplicaciones y entornos que dejan de lado a este particular teléfono inteligente. Pero un poco por ser el que dio el puntapié en la era de los smartphones, por sostenerse en el mercado y porque hay tela para cortar aún le dedicaremos un capítulo con una guía para desarrollar aplicaciones. A lo largo del artículo veremos que la plataforma web de BlackBerry nos proporciona los elementos básicos para trabajar sin necesidad de conocimientos específicos de programación. Inclusive, una vez creada una API podremos probarla en un simulador antes de ponerla en funcionamiento. Recuerden que RIM es una compañía muy estricta en cuanto a los derechos de apli-

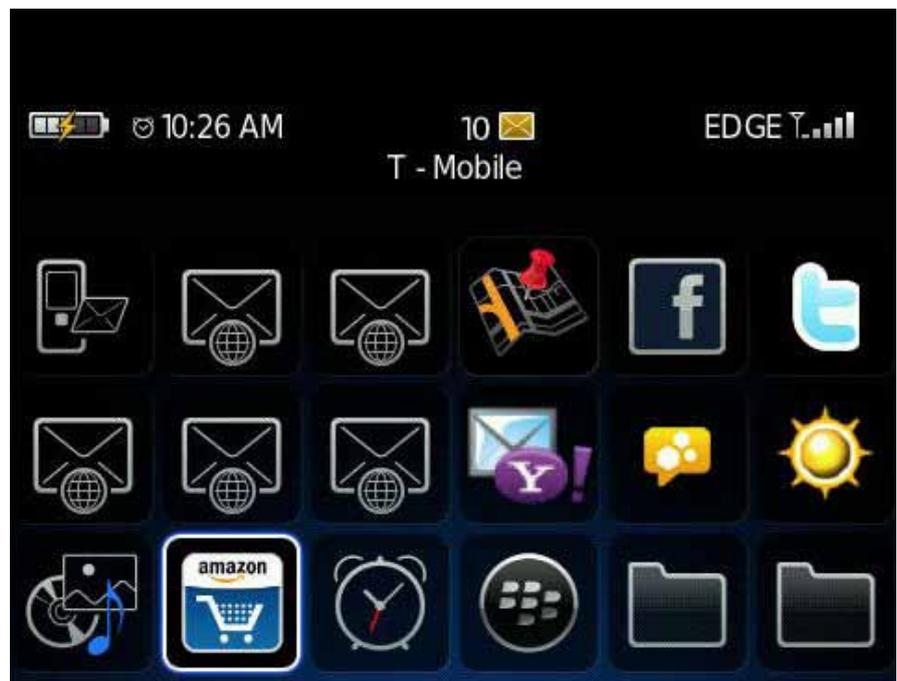
caciones, por lo que cada creación deberá llevar una firma que certifique su uso.

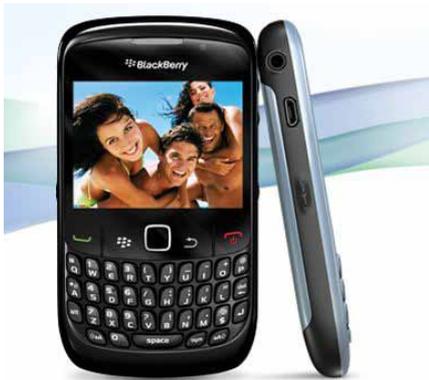
### **REQUISITOS ANTES DE EMPEZAR**

Como lo mencionamos anteriormente, no hay que ser un programador experto pero cierto es que si tenemos un manejo mínimo de ciertos conceptos, además del uso del dispositivo inteligente, nos servirá para los fines de poner en práctica algunas ideas sabiendo con qué recursos contamos. De esa manera no perderemos tiempo

solucionando errores. Otra opción es recurrir a los cientos de tutoriales que ofrece la compañía, pero para eso estamos puestos a hacer de este artículo una guía que nos simplifiquen esos pasos.

En principio necesitaremos instalar el JDE de BlackBerry, que es el entorno de desarrollo completo que se usa para escribir el código. También se puede utilizar un IDE (entorno de desarrollo integrado) que es un programa informático que contiene un grupo de herramientas de programación. También será necesario descargar una serie de archivos para el desarrollo en





*“No hay que ser un programador experto pero cierto es que si tenemos un manejo mínimo de ciertos conceptos, además del uso del dispositivo inteligente, nos servirá para los fines de poner en práctica algunas ideas sabiendo con qué recursos contamos.”*

Java y el archivo BlackBerry Java Development Environment v4.2.1 que proporcionará herramientas IDE para desarrollar aplicaciones, para emular un dispositivo y para depurar la aplicación en un equipo real o en el emulador, además de optimizadores de memoria, uso de threads y más.

Si no quieres acceder a un BlackBerry Enterprise Server, entonces puedes usar MDS Studio que no es Java ME pero es similar a Javascript con XML con un buen entorno visual. Por su parte, JDE exige obtener algunas herramientas para su instalación y uso como Java SE Development Kit versión 1.6 o superior, Microsoft DirectX version 8.0 o posterior, Microsoft Windows Vista o Microsoft Windows XP Professional, procesador Intel Pentium o compatible (800 MHz o superior)

un mínimo de 400 MB de Ram disponible o más y un mínimo de 500 MB de espacio en el disco duro disponible.

En caso de crear desarrollos para BlackBerry de código abierto, será necesario contar con un poco de experiencia en desarrollo de aplicaciones para móviles, algunas habilidades de programación Java, PHP y MySQL.

### **FIRMAS PARA CUIDADOS INTENSIVOS**

A la hora de ganar un espacio para crear aplicaciones y ponerlas a disposición de los distintos usuarios del dispositivo, uno de los temores comunes es que estos desarrollos contengan algunas aplicaciones un tanto maliciosas que oculten un rastreo de datos o usurpación oculta de información del usuario. Por eso Research In Motion hace un seguimiento de las interfaces de programas de aplicación de BlackBerry controlando la seguridad que ofrecen a sus clientes.

Tanto para desarrollar como para investigar en profundidad te encontrarás con un gran número de APIs que están firmadas con una clave, esta firma la proporciona RIM para autentificar el uso correcto. Si no puedes cargar la aplicación o da un error “”, entonces esa API no está firmada.

Esta firma sólo es solicitada cuando subimos la aplicación al





dispositivo para ponerla a prueba, aunque no es solicitada cuando se prueba en un simulador. El desarrollador deberá abonar 20 dólares por pago electrónico y completar un formulario que solicita un PIN necesario para instalar las Keys. RIM enviará una información en tres correos con archivos adjuntos (RBB, RCR, RRT) que hay que descargar y ejecutar en donde llevemos a cabo el desarrollo y esa firma sólo servirá para esa PC. Deberán identificar un password para las tres keys y que no es el PIN antes mencionado. Una vez con estos condimentos en la cacerola sólo basta ponernos a cocinar y para ello, cabe aclarar, que se dispondrá de ejemplos de aplicaciones realizadas por otros desarrolladores con el fin de ilustrar el artículo.

## **DESARROLLAR UNA APLICACIÓN**

Veremos cómo desarrollar un

acceso directo a un sitio móvil, lo que también se denomina Web Launcher. Tengan presente que se utilizarán nombres de carpetas elegidas por los desarrolladores consultados y que esto no significa que siempre se generen esas mismas estructuras. Dependerá de cada uno el cómo organiza sus proyectos. En este caso trabajaremos con un ejemplo utilizado por Javier González para el sitio ZonaBlackBerry.com.ve.

Como primera medida, colocaremos cada proyecto JDE en una carpeta diferente de otra. De la misma se desprende la carpeta "com" que contiene el proyecto bajo la extensión .jdp, también allí encontraremos la imagen o ícono .png que identifique nuestra aplicación con la extensión pero dentro de una carpeta llamada "jmg". En otra carpeta se guardará el archivo de extensión java. Haciendo click derecho sobre la carpeta principal podremos ver las propiedades de la misma a la dere-

cha de la pantalla.

Se abrirá una opción de propiedades con distintas pestañas donde agregaremos información: General, pondremos el título de la aplicación, versión, propietario y descripción. Application, especificaremos el tipo de proyecto. Resources, agregaremos las imágenes o recursos de la aplicación. Con estos pasos hemos definido las propiedades del proyecto, ahora sólo hay que desarrollar el código fuente o source code, que en el caso de este ejemplo es:

```
/*
 * ZonaBlackberry.java
 * © <ZonaBlackberry>,
 2008-2009
 * by Javier Gonzalez
 */
package com.ZonaBlackberry;
import net.rim.blackberry.api.browser.Browser;
import net.rim.blackberry.api.homescreen.HomeS-
```

## >PROGRAMACIÓN PARA MÓVILES-



“El desarrollador deberá abonar 20 dólares por pago electrónico y completar un formulario que solicita un PIN necesario para instalar las Keys.”

```
creen;  
import net.rim.device.  
api.system.Application;  
import net.rim.device.  
api.system.Bitmap;
```

```
public class Zona Black-  
berry extends Application  
{  
  
    public static void  
main (String [ ] args)  
    {  
        Browser.ge-  
tDefaultSession().  
displayPage (“http://zona-  
blackberry.com.ve”);  
    }  
}
```

Una vez obtenido el código hay que probarlo en un simulador. Esta versión dependerá del JDE instalado para trabajar en el desarrollo. Si descartamos cualquier fallo,

entonces estamos listos para firmar la aplicación y construir el proyecto que dará origen a los archivos .cod, .jar y .jad. Una vez que obtenemos los keys para la firma ingresamos por Build/Request Signatures donde validaremos las claves. Este paso requiere de conexión a Internet. El paso final es generar el archivo .alx para instalar vía Desktop Manager. Una vez con todos los archivos en la carpeta “com”, los podemos copiar para subirlos a un servidor de descarga OTA, que es un servidor de descarga directa. Finalmente, veremos en nuestro BlackBerry el ícono en la pantalla que nos introduce en la aplicación que acabamos de crear.

### **PAQUETE DEL ENTORNO RIM**

La compañía ofrece a los desarrolla-

dores un buen grupo de herramientas para trabajar sus propias creaciones con libertad. Estas incluyen, además de un editor de código gráfico e instrumentos de complicación, unos cuantos recursos más como simuladores, depuradores y guías de buenas prácticas de codificación. Simuladores: Puestos a trabajar nadie quiere perder tiempo buscando el error o falla que no permite que nuestro proyecto se lleve a cabo, para ello, RIM ofrece distintas instancias de simulación que incluyen desde la prueba del proyecto en sí, del tráfico de comunicaciones entre un BlackBerry real y la red de RIM, y un destacado simulador de email. Siendo que este es el fuerte de los BlackBerry, no es menor poner a prueba aplicaciones relacionadas con la funcionalidad que nos evitarán además de dolores de cabeza la decepción de los clientes más asiduos de BB.

### ESQUEMA DE TRABAJO

Sea cual fuere nuestro desarrollo y para los fines organizativos, conviene esquematizar nuestra aplicación teniendo bien claro qué queremos lograr con ella y a cuántos usuarios les será útil; de esta forma, lograremos diseñar un alcance más completo sin que esto se nos convierta en un propósito completo. Hemos visto que desarrollar

una aplicación es más que sencillo, ahora bien, que esta sea utilizada por los usuarios puede resultar una tarea complicada si la aplicación que ideamos no tiene parámetros del todo útiles o prácticos. Recordemos que los usuarios de BlackBerry recurren al mismo por su acceso al correo electrónico, las opciones de comunicación instantánea y la practicidad que ofrece para llevar el escritorio de la oficina en sus manos. Con estas observaciones nos será más fácil plantear proyectos no tan fascinantes pero sí sencillamente lucrativos.

La estructura de nuestra aplicación comienza entonces por un punto de entrada o principal al que luego le daremos la propiedad de aplicación generando una primera instancia de “demostración”. A medida que depuramos esta demostración lograremos un desarrollo cada vez más pulido y libre de errores. Hasta alcanzar el archivo que instalaremos en la instancia final y que pondrá sobre el tapete de nuestro smartphone cuán capaces somos de crear un desarrollo inteligente para un dispositivo inteligente. **P**





# Atributos decorativos y Reflection

Por Matías Iacono  
matias.iacono@pixelscode.com

Los marcos de desarrollo de software, en especial aquellos que soportan código intermedio o código manejado, como Microsoft .Net o Java, así como otros de alto nivel cuentan con una cualidad interesante, pero a veces poco usada. Esta cualidad, llamada Reflection (también podemos encontrarlo en castellano como introspección o reflexión) les permite a los desarrolladores de software que trabajan con estas plataformas ver, analizar y modificar el código desde el mismo código en el momento en que éste se ejecuta.

Si pensamos en algunas de las posibilidades de implementación que esta técnica nos puede dar, rápidamente podríamos plantear la del análisis de métodos de un objeto y ejecutar todos sin que sepamos, realmente, cuáles o cuántos existen; otra posibilidad podría ser la de agrupar y seleccionar sólo aquellos objetos que cumplan con una determinada característica como, por ejemplo, implementar una interfaz específica, heredar de una clase base particular, o contener un constructor con un parámetro especial. En todo caso, las posibilidades son variadas, pero posiblemente lo más importante es que nos fuerzan a ir un paso más lejos dentro del desarrollo de las líneas de código y pensar estas más allá de una clásica implementación de los conceptos de orientación a objeto. Para entender esto último, imaginemos el escenario clásico en una aplicación de orientación a objetos. Un conjunto de clases especializadas, objetos que son pasados entre ellas, y cada una realizando acciones específicas sobre estos objetos para luego desencadenar una acción. Este escenario podría aplicarse a la ejecución de reglas de validación de contraseñas, y posiblemente podríamos aplicar un patrón clásico de cadena de responsabilidades para el encadenado y la posterior ejecución de cada una de las reglas. Veamos una implementación de este patrón.

```
internal interface IEslabon
{
    void Ejecutar(string valor);
    IEslabon SiguinteEslabon { get; set; }
}
```

Como primer elemento, creamos una interfaz de la cual los eslabones de la cadena de responsabilidades implementarán. De esta forma, cada clase sólo necesitará ser parte de este grupo para poder ser incluida dentro de la cadena.

```

internal class Eslavon1 : IEslabon
{
    public void Ejecutar(string valor)
    {
        if (valor.Length <= 7)
            Console.WriteLine("Cadena muy corta");
        else
            if (SiguienteEslabon != null)
                SiguienteEslabon.Ejecutar(valor);
    }
    public IEslabon SiguienteEslabon { get; set; }
}

```

```

internal class Eslavon2 : IEslabon
{
    public void Ejecutar(string valor)
    {
        if (valor.Length >= 15)
            Console.WriteLine("Cadena muy larga");
        else
            if (SiguienteEslabon != null)
                SiguienteEslabon.Ejecutar(valor);
    }
    public IEslabon SiguienteEslabon { get; set; }
}

```

Como vemos en el código anterior, hemos creado dos clases que implementan de IEslabon, y cada una de estas se especializa en un tipo de validación del texto. Si la condición se cumple, un mensaje es mostrado en la consola de comandos y si no, se verifica la existencia de un siguiente eslabón en esta cadena para ejecutarlo y así continuar con la validación del texto.

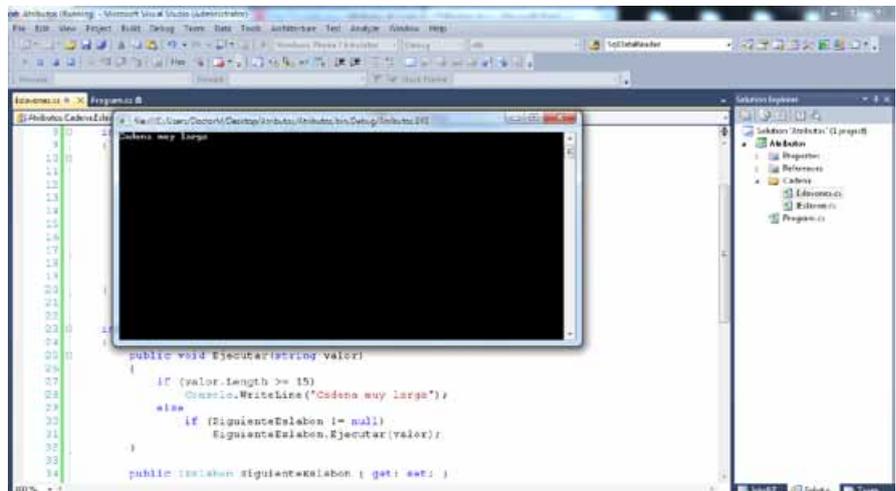
Al ejecutar la cadena de responsabilidades, la regla del largo se cumple y un mensaje se muestra en la consola de comandos.

Indudablemente, si necesitamos otro eslabón que aplique otra regla de validación, sólo tendremos que crear una nueva clase, implementar de IEslabon y adicionar el código para realizar la validación. En todo caso, un posible problema se presenta en el momento de la creación de la cadena en sí.

```

static void Main(string[]
args)
{

```



*“ Si pensamos en algunas de las posibilidades de implementación que esta técnica nos puede dar, rápidamente podríamos plantear la del análisis de métodos de un objeto y ejecutar todos sin que sepamos, realmente, cuáles o cuántos existen.”*

## >BASES DE DATOS (II)-

```
IEslabon eslavon1 = new Eslavon1();
IEslabon eslavon2 = new Eslavon2();

eslavon1.SiguienteEslabon = eslavon2;
eslavon2.SiguienteEslabon = null;

eslavon1.Ejecutar("Texto a validar");

Console.ReadLine();
}
```

Por cada eslabón de la cadena es necesario crear código que lo una al anterior eslabón en la cadena. Esto puede ser un problema si nos viéramos en la necesidad de agregar o quitar eslabones de forma dinámica, o si los eslabones provinieran de librerías de código externas al programa principal. Si bien es necesario destacar que el patrón propuesto es sólido, y el problema presentado puede no ser tal en la mayoría de los casos, lo proponemos como ejemplo por su simpleza.

### **CREANDO LA CADENA CON REFLECTION**

Teniendo el dilema anterior, un acercamiento a una solución diferente podría ser la de utilizar lo que planteábamos al principio de este artículo. Mediante Reflection, recorrer todos los tipos contenidos dentro del ensamblado de nuestra aplicación, analizar cuáles de ellos cumplen con la condición de ser eslabones de nuestra cadena y crear la cadena de forma dinámica. Para esto crearemos un objeto que actuará de fábrica o factoría de la cadena.

```
public class FactoriaCadena
{
    private IEslabon eslabonInicial;
    private IEslabon siguienteEslabon;
    public FactoriaCadena()
    {
        Type[] types = Assembly.GetAssembly(typeof(IEslabon)).GetTypes();
        foreach (Type type in types)
        {
            if (!typeof(IEslabon).IsAssignableFrom(type)
                || type == typeof(IEslabon))
                continue;
            if (eslabonInicial == null)
            {
                eslabonInicial = (IEslabon)Activator.CreateInstance(type);
                siguienteEslabon = eslabonInicial;
            }
            else
            {
                IEslabon eslabonTemporal = (IEslabon)Activator.
CreateInstance(type);
                siguienteEslabon.SiguienteEslabon = eslabonTemporal;
                siguienteEslabon = eslabonTemporal;
            }
        }
    }
}
```

```

public void Procesar(string valor)
{
    eslabonInicial.Ejecutar(valor);
}
}

```

Como vemos en el código anterior, Microsoft .Net y C# presentan unas líneas particulares para realizar esta tarea. Primero, la inspección de todos los tipos (Clases, interfaces, etc.) que cumplan con la condición de implementar la interfaz IEslabon. Interfaz que utilizamos inicialmente para la formación de la cadena de responsabilidades.

```

Type[] types = Assembly.GetAssembly(typeof(IEslabon)).GetTypes();
foreach (Type type in types)
{
    if (!typeof(IEslabon).IsAssignableFrom(type)
        || type == typeof(IEslabon))
        continue;
}

```

La directiva Assembly nos permite interactuar con los paquetes de código creados por Microsoft .Net (Assembly o ensamblado es el nombre que reciben nuestras aplicaciones en esta tecnología, ya sean archivos con extensión DLL o EXE), el siguiente paso será crear nuevas instancias de estos tipos encontrados y armar la cadena de forma dinámica para su posterior ejecución. La siguiente línea se encarga de crear esta instancia en base al tipo definido.

```

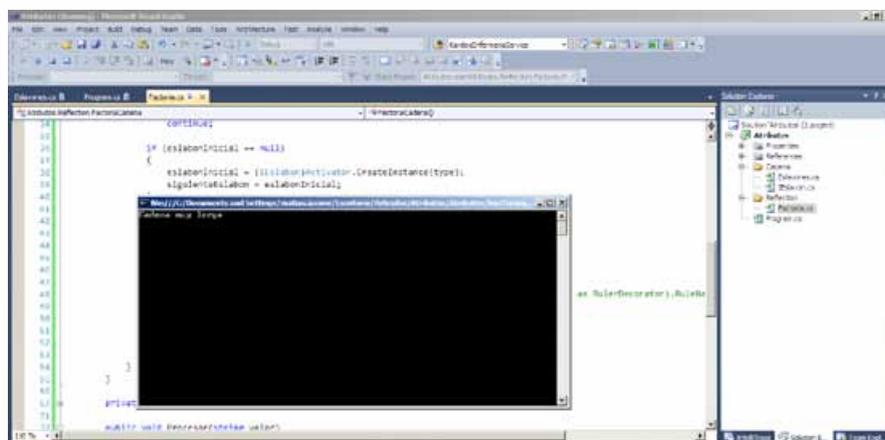
eslabonInicial = (IEslabon)Activator.CreateInstance(type);

```

Al ejecutar la cadena de responsabilidades desde la factoría el resultado obtenido es el mismo que el código anterior. La ventaja significativa de este método es que cualquier nueva clase que sea adicionada a nuestro código y que implemente la interfaz IEslabon, automáticamente pasará a ser parte de la cadena, lo que nos ahorrará tiempo de desarrollo y mantenimiento para sobre las líneas que construyan la cadena en sí. En todo caso, podemos llevar este ejemplo un paso más lejos.

## ATRIBUTOS DECORATIVOS

Los atributos decorativos se caracterizan, o pueden ser detectados en la programación con C# y Microsoft .Net, por etiquetas descriptivas colocadas en la cabecera de las clases, funciones o propiedades. Estas etiquetas son a su vez código estático que acompaña a la descripción general de aquello que describe. Por lo tanto, podemos, mediante Reflection, analizar el contenido de esas descripciones y actuar en consecuencia. Imaginemos que, a pesar de la flexibilidad del código propuesto anteriormente, aún necesitamos realizar algunos cambios en nuestro código para poder modificar los estados de estas cadenas de responsabilidad. Esto es, necesitaremos recompilar el código si queremos hacer algún cambio. De la misma forma, no podremos elegir cómo queremos que esta cadena se arme, qué eslabón deba ir después del otro, lo que nos llevaría a que la mejor solución sea volver al patrón original. En todo caso, podríamos plantearnos que, mediante Reflec-



## >BASES DE DATOS (II)-

tion, atributos decorativos y algún archivo externo podamos configurar todo esto sin la necesidad de tener que modificar el código interno de nuestra aplicación, que a su vez, ya está compilado. Primero, entonces, definamos una clase que actuará como decorador de las demás clases.

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false)]
internal sealed class Decorador
    : Attribute
{
    public Decorador(string nodo)
    {
        this.Nodo = nodo;
    }
    public string Nodo { get; internal set; }
}
```

Como podemos observar, esta clase además de heredar de la clase Attribute (Atributo), también implementa atributos decorativos para especificar su comportamiento. En este caso, se especifica que esta clase, como un atributo decorativo, será aplicable sólo a otras clases y que no permitirá múltiples asignaciones, o sea, no podrá encontrarse más de un atributo de este tipo por clase. Hagamos ahora las modificaciones necesarias a las clases que representan cada eslabón para que usen este atributo.

```
[Decorador("Eslabon1")]
internal class Eslavon1 : IEslabon
{ ... }
[Decorador("Eslabon2")]
internal class Eslavon2 : IEslabon
{ ... }
```

Cada eslabón ahora se describe con el atributo decorativo y especifica un nombre representante dentro de la cadena. De esta forma, podríamos tener un archivo XML que represente grupos de cadenas y qué eslabones encontraremos dentro de cada uno.

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuracion>
  <Grupo Nombre="Cadena1">
    <Eslabon Nombre="Eslabon1" />
    <Eslabon Nombre="Eslabon2" />
  </Grupo>
</Configuracion>
```

Así, cada grupo podrá agregar, quitar o modificar el orden de creación de la cadena sin necesidad de recompilar el código, sino sólo modificando este archivo XML externo. Por supuesto, necesitaremos modificar un poco nuestra factoría para que ahora tome los valores y el orden desde estos atributos decorativos y el archivo XML.

```
public FactoriaCadena(string grupo)
{
```



*(...) un gestor de datos no necesitaría conocer un tipo de objeto con valores, sólo saber qué propiedades están decoradas y actuar en consecuencia insertando, eliminando o actualizando esos datos."*

```

List<string> eslabones = ObtenerListaEslabones(grupo);

Type[] types = Assembly.GetAssembly(typeof(IEslabon)).GetTypes();
foreach (Type type in types)
{
    ...
    ...
    string nombreNodo = (type.GetCustomAttributes(typeof(Decorador), true)[0]
as Decorador).Nodo;
    if (eslabones.Contains(nombreNodo))
    {
        ...
        ...
    }
}
}

```

En el código hemos usado una función auxiliar que retorna la lista de eslabones de acuerdo al archivo XML. A continuación podemos ver esta función.

```

private static List<string> ObtenerListaEslabones(string grupo)
{
    XmlDocument xDocumento = new XmlDocument();
    xDocumento.Load(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "ConfiguradorCadena.xml"));
    XmlNodeList xLista = xDocumento.SelectNodes("//Configuracion/Grupo[@Nombre=' " + grupo + "' ]/Eslabon");

    List<string> listaNodos = new List<string>();
    foreach (XmlNode item in xLista)
    {
        listaNodos.Add(item.Attributes["Nombre"].Value);
    }
    return listaNodos;
}

```

Si ejecutamos el código ahora podremos comprobar que la cadena seguirá comportándose de forma esperada con la diferencia de que podremos modificar su comportamiento sólo modificando el archivo XML gracias a la inspección que realiza la factoría sobre los atributos decorativos de cada clase.

Los atributos decorativos no sólo sirven para este tipo de implementaciones. Imaginemos que muchos ORMs y marcos de trabajo para el acceso a datos, entre muchos otros, utilizan este tipo de características para aislar el objeto que contiene una funcionalidad específica de la dependencia hacia el negocio o la ejecución de reglas lo que hace a nuestro código, en algún punto, desacoplarse y trabajar de forma independiente. Así, un gestor de datos no necesitaría conocer un tipo de objeto con valores, sólo saber qué propiedades están decoradas y actuar en consecuencia insertando, eliminando o actualizando esos datos, o seleccionándolos de una base de datos y rellenando el objeto entidad de forma automática sin importar, en definitiva, de qué objeto se trate. **P**

# Email Marketing con tu marca

Ganá dinero ofreciendo a tus  
clientes la mejor herramienta.

