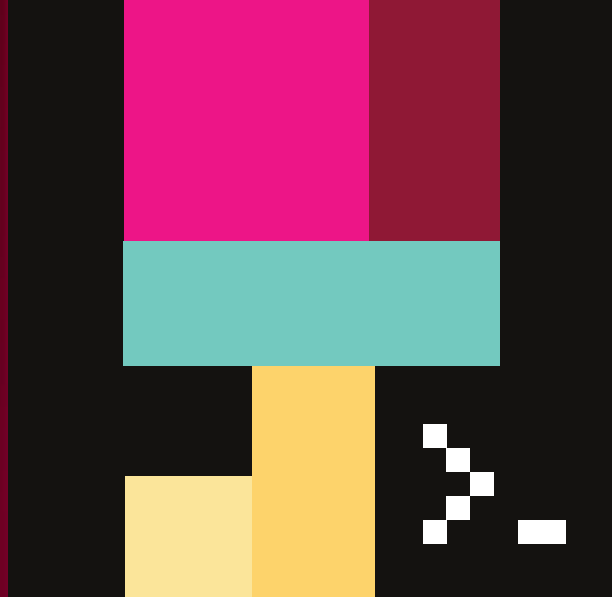




Pixels&Code

ENERO 2012 > N° 09 -

**TODO SOBRE
ANDROID 4.0**

**DART, ¿EL NUEVO
JAVASCRIPT?**

JUGANDO CON CSS3

**TRUCOS PARA LOGRAR TODO TIPO DE EFECTOS EN TUS
PÁGINAS, SIN RECURRIR A PROGRAMAS DE DISEÑO.**

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



Con .TV la gente te ve.

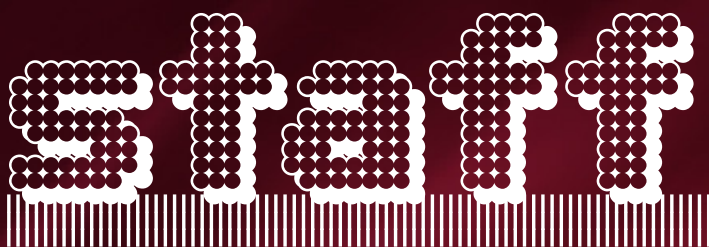
Si tenés **videos**,
la mejor forma de mostrarlos
es con un **dominio .tv**

Entrá a **www.lagente.tv** y registrá el tuyo.



dattatec.com

Los especialistas en dominios



dattatec.com
Tu Hosting hecho Simple!

DIRECTORA Y JEFA DE REDACCIÓN



Débora Orué

COLUMNISTAS

*Fernando Luna
Gabriel Walter Gaitano Ornia
Juan Gutmann*

*Mariano Salvetti
Matías Iácono
Roxana Miguel*

DISEÑO EDITORIAL Y CREATIVIDAD

www.ampersandgroup.com.ar

REDACCIÓN

lectores@pixelscode.com

COMERCIALIZACIÓN

*Dattatec.com SRL - Córdoba 3753, Rosario, Santa Fe
www.dattatec.com*

DATOS DE CONTACTO

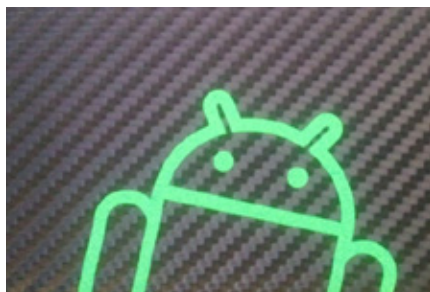
*Dirección Comercial:
publicidad@pixelscode.com*

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

sumario

NOTA CENTRAL

JUGANDO CON CSS3



6 // INFORMACIÓN GENERAL

F5: Entry point

8 // BASES DE DATOS

Introducción a las bases de datos relacionales (IX)

18 // DESARROLLO PARA MÓVILES

Novedades en Android para Developers

24 // LENGUAJES

Google Dart: El nuevo JavaScript

36 // PROGRAMACIÓN

Código dinámico con T4: Segunda parte

42 // INFORMACIÓN GENERAL

Coworking: Tu free-office

48 // ALGORITMOS

Algoritmos: comb sort

54 // WEBMASTERS

Optimiza tu site con Google Webmaster Tools



F5: Entry point

Por Gabriel Walter Gaitano Ornia

gabriel.gaitanoornia@pixelscode.com



WEB PARA CONOCER: **SQLSERVERCENTRAL.COM**

Entre los softwares de gestión de bases de datos más usados se cuenta, sin duda, el conocido SQL Server de Microsoft. Para todos aquellos que lo utilizan, sea directamente como administradores de base de datos o como programadores que acceden a los datos desde su código, les resultará excepcionalmente interesante la muy completa web

<http://www.sqlservercentral.com>

Plena de completos artículos, foros de consulta e informes sobre todos los temas relacionados con el producto y, en un sentido más amplio, sobre el propio lenguaje de consulta

SQL, la web es útil para distintos niveles de usuarios incluyendo a los expertos. Tras suscribirnos, un completo newsletter nos permitirá recibir referencia a las novedades del sitio de modo periódico, aun si no recordamos visitarlo. Realmente una buena página que no podemos ignorar si el SQL es nuestra herramienta de trabajo.

GADGETS: **MOTOROLA DROID RAZR**

Considerado el celular más delgado del mundo, incluye un procesador de doble núcleo a 1,2 GHZ, 1 GB de RAM y Android 2.3.5 upgradeable al novísimo Android 4, Ice Cream Sandwich ("Casatta" para nosotros,

los argentinos). Está blindado en kevlar para hacerlo más resistente, el mismo material del que se hacen los chalecos antibalas y el traje ultra-resistente de Batman. Trae cámara de 8 MP para una alta definición y pantalla de 4.3". Pesa apenas 134 gramos.

Todo esto y más sirve para definir la última creación de los diseñadores de Motorola, el Droid RAZR. Podemos ver sus características y detalles desde <http://bit.ly/nQKDL1>. Personal ya lo comercializa en Argentina, exclusivamente con su plan Personal Black, de \$199 por mes. Todo un objeto de deseo con forma de smartphone.



IT NEWS: **FIREFOX VS. CHROME**

¿Se avecinan nubarrones para Firefox? Un reporte muy reciente de la consultora StatCounter señaló que finalmente el navegador de Google, Chrome, logró ubicarse detrás del ubicuo Internet Explorer, quien aún lidera con un 40.63% del mercado, colocándose de este modo ligeramente por delante de Firefox. Aunque el margen es muy reducido, queda claro que Chrome ha ganado un notable porcentaje de mercado mientras su rival se ha estancado. Sin un adecuado sistema de aplicaciones y habiendo sufrido muchas quejas por un excesivo consumo de recursos, Firefox deberá hacer un notable esfuerzo para detener la marea en su contra.

MÁS ALLÁ DE LA PC: **HP LIBERÓ EL CÓDIGO** **DE WebOS**

Un último intento antes del final: la compañía Hewlett-Packard decidió, ante la imposibilidad de seguir apostando a su desarrollo tras abandonar el mercado de tablets y celulares, liberar el código fuente del sistema operativo móvil WebOS. De acuerdo a lo que se comenta, puede haber

dos lecturas para esta noticia: o bien HP no pudo encontrar un comprador razonable, o está apostando a que pueda lograr abrirse camino por sí mismo en el mundo del open source.

De este modo, el sistema operativo que fuera insignia de la nave de HP desde la adquisición de Palm es abandonado, tras haberse transformado en fuente de pérdidas que invalidaron las grandes expectativas iniciales.

Con este movimiento, la compañía espera que el WebOS reviva de la mano de una comunidad de usuarios deseosos de mantenerlo vivo. Y, en caso de derrumbarse definitivamente en una tumba de código libre, el impacto, estiman en HP, será mucho menor que manteniéndolo en sus manos.

SEGURIDAD: FACEBOOK **Y LAS FOTOS PRIVADAS** **DE ZUCKERBERG**

En medio de un creciente cuestionamiento público a los lineamientos de seguridad y corporativos para preservar los datos personales de los usuarios de Facebook, varias fotos privadas del fundador y actual líder de la red social fueron reveladas en

Internet, muy a su pesar. Luego pudo llegar a saberse que la falla había afectado al universo completo de usuarios de la red.

¿Cómo pudo ser esto posible? Al parecer, fue tan sencillo como utilizar un truco de reciente publicación en Internet, el cual permitía acceder a las fotos marcadas como privadas si uno fraguaba una “denuncia” de una foto disponible del usuario del sitio. Así era posible, involuntariamente, denunciar otras imágenes incluidas en ese perfil en cuestión, incluso aquellas que se debían mantener privadas. Esto, naturalmente, podía ser aprovechado con otros fines y, a pesar de no cumplirse siempre el efecto, funcionó. De hecho, funcionó tan bien que permitió a varios usuarios curiosos aprovechar la oportunidad para ingresar en el perfil de su fundador, Mark Zuckerberg.

Así fue que se pudieron ver fotos suyas en situaciones domésticas, como ser cocinando o sosteniendo sonriente un pollo ante la cámara. **P**



Introducción a las bases de datos relacionales (IX)

Por Juan Gutmann
juan.gutmann@pixelscode.com

Como ya hemos visto en entregas anteriores, los Triggers (conocidos en español como “disparadores”) son muy útiles para otorgar a nuestra base de datos algunas funcionalidades interesantes con muy poco trabajo, como la generación automática de claves, o la implementación de sofisticadas reglas de integridad referencial y lógica de negocios. Una característica fundamental que deben poseer las bases de datos en ciertos dominios, como el mundo bancario o financiero, es poder auditar en detalle cómo ha ido evolucionando la información contenida en el motor relacional. Es decir, no solamente es importante almacenar los datos y mantener la consistencia de los mismos, sino que también es importantísimo saber con precisión el usuario y la fecha exactas en la que se creó, modificó o eliminó un dato determinado. Gracias a los triggers, podemos implementar una solución completa de auditoría en poco tiempo. Aquí vamos a demostrar cómo hacerlo con PostGres gracias a las posibilidades de PL/pgSQL.

DOS ENFOQUES DISTINTOS

Como en casi todo problema que pueda solucionarse mediante un algoritmo escrito en cualquier lenguaje de programación, existen diversas formas de resolverlo, cada una con sus particularidades, ventajas e inconvenientes. Plantearemos dos maneras de incorporar un exhaustivo registro de auditoría a un modelo de datos. Ambas son relativamente populares, aunque tienen enfoques completamente diferentes. Han sido armadas sobre la base del modelo de datos de práctica que venimos desarrollando a lo largo de esta serie de notas, y aunque la demostración de cada una de ellas trabaja sobre una tabla sola (empleado), es extremadamente simple aplicar la misma lógica a todas las entidades del modelo que se desee auditar.

PRIMERA SOLUCIÓN: FÁCIL Y ELEGANTE

En la primera alternativa que proponemos, se privilegia la escritura de código compacto, sencillo de entender y modificar. Otra virtud de esta versión es que las tablas de auditoría resultantes son extremadamente fáciles de consultar,



ya que tienen una estructura casi idéntica a cada una de las tablas auditadas. Obviamente, en cada tabla histórica se almacenarán no solamente los datos que componen cada registro tras un cambio, sino que también se le agregará la información necesaria para auditar cómo sucedió el mismo, es decir, el usuario que lo realizó y la fecha y hora en la que se produjo. Las contras de esta modalidad de auditoría son dos. En primer lugar, se requiere de una tabla, un procedimiento almacenado y un trigger para cada una de las tablas que se quiere auditar. Además, ante cada alta, modificación o baja de un registro en dicha tabla, se guarda una copia completa del nuevo estado del registro, aunque se haya modificado apenas un campo del mismo. Esto es muy poco óptimo en lo que a aprovechamiento del espacio en la base de datos se refiere, lo cual puede transformarse en una limitación crítica bajo determinadas circunstancias. Pasemos a ver cómo está armada esta implementación:

```
CREATE TABLE empleado_historico (
accion          VARCHAR(10)    NOT NULL,
fecha           TIMESTAMP      DEFAULT now() NOT NULL,
usuario         VARCHAR(63)    DEFAULT user NOT NULL,
legajo          NUMERIC(5)     NOT NULL,
cargo           VARCHAR(10)    NOT NULL,
apellido_y_nombre VARCHAR(100) NOT NULL,
supervisor      NUMERIC(5)     NOT NULL,
CONSTRAINT pk_empleado_historico PRIMARY KEY (fecha, legajo)
);
```

Comenzamos creando una tabla que llamamos “empleado_historico”, con idéntica estructura a la tabla a auditar -en este caso, “empleado”- incorporando tres campos más: “accion” (en el cual registraremos si se está creando, modificando o eliminando el registro), “fecha” (donde se guardará la fecha y hora en la que tuvo lugar el cambio) y “usuario” (que almacenará el nombre del usuario que realizó la operación). Para asignar una clave única a la misma no alcanza con recurrir al mismo campo que la tabla original (en este caso, legajo), ya que para cada registro de dicha tabla tendremos en el histórico tantas filas como cambios haya sufrido aquel. Una virtud de esta estructura de la tabla histórica es que no requiere la creación de una clave artificial especialmente para cada registro, sino que la “clave candidata” que podemos tomar para su empleo como clave única está compuesta por la clave original de la tabla a auditar, con el agregado del atributo “fecha”, ya que no pueden producirse simultáneamente dos modificaciones distintas en un mismo registro.

>BASES DE DATOS-

“En la primera alternativa que proponemos, se privilegia la escritura de código compacto, sencillo de entender y modificar.”

“El “Trigger Procedure” (el procedimiento almacenado que será invocado por el trigger que se dispara al producirse cualquier evento sobre la tabla “empleado”) es similar a los que ya hemos visto, pero se vale de una de las variables internas que PostGres provee especialmente a los Trigger Procedures.”

```
CREATE FUNCTION f_trg_empleado_historico() RETURNS
trigger AS $$
BEGIN
    IF TG_OP = 'INSERT' OR TG_OP = 'UPDATE' THEN
        INSERT INTO empleado_historico
            (accion, legajo, cargo, apellido_y_nombre, supervisor)
            SELECT TG_OP, NEW.*;
        RETURN NEW;
    END IF;

    IF TG_OP = 'DELETE' THEN
        INSERT INTO empleado_historico
            (accion, legajo, cargo, apellido_y_nombre, supervisor)
            SELECT TG_OP, OLD.*;
        RETURN OLD;
    END IF;

END;
$$ LANGUAGE plpgsql;
```

El “Trigger Procedure” (el procedimiento almacenado que será invocado por el trigger que se dispara al producirse cualquier evento sobre la tabla “empleado”) es similar a los que ya hemos visto, pero se vale de una de las variables internas que PostGres provee especialmente a los Trigger Procedures. Ya conocemos bien “NEW” y “OLD”, que contienen copias de las versiones

```
1 CREATE SEQUENCE seq_historico
2 INCREMENT BY 1
3 NO MINVALUE
4 NO MAXVALUE
5 START WITH 1
6 CACHE 1
7 NO CYCLE
8 OWNED BY NONE;
9
10 CREATE TABLE historico (
11     clave          NUMERIC(10)  DEFAULT nextval('seq_historico') NOT NULL,
12     tabla          VARCHAR(63)  NOT NULL,
13     accion         VARCHAR(10)  NOT NULL,
14     campo          VARCHAR(63)  NOT NULL,
15     usuario        VARCHAR(63)  DEFAULT user NOT NULL,
16     fecha          TIMESTAMP    DEFAULT now() NOT NULL,
17     dato_viejo     VARCHAR(4000),
18     dato_nuevo     VARCHAR(4000),
19     CONSTRAINT pk_historico PRIMARY KEY (clave)
20 );
21
22 CREATE FUNCTION f_trg_historico() RETURNS trigger AS $$
23     DECLARE tcampo_new TEXT;
24     DECLARE tcampo_old TEXT;
25
```



```

C:\WINDOWS\system32\cmd.exe - psql -U postgres
practicass=# INSERT INTO empleado
practicass=# (cargo, apellido_y_nombre, supervisor)
practicass=# VALUES
practicass=# ('ACMIVO', 'DOMINGUEZ ALEJANDRO', 2);
INSERT 0 1
practicass=# UPDATE empleado
practicass=# SET apellido_y_nombre = 'DOMINGUEZ ALEJANDRO DAMIAN',
practicass=# supervisor = 2000
practicass=# WHERE legajo = 2001;
UPDATE 1
practicass=# DELETE FROM empleado
practicass=# WHERE legajo = 2001;
DELETE 1
practicass=# SELECT * FROM empleado_historico;
 action | fecha          | usuario | legajo | cargo | apellido_y_nombre | supervisor
-----|-----|-----|-----|-----|-----|-----
 INSERT | 2011-12-12 21:22:40.171 | postgres | 2001 | ACMIVO | DOMINGUEZ ALEJANDRO | 2
 UPDATE | 2011-12-12 21:22:47.468 | postgres | 2001 | ACMIVO | DOMINGUEZ ALEJANDRO DAMIAN | 2000
 DELETE | 2011-12-12 21:22:54.281 | postgres | 2001 | ACMIVO | DOMINGUEZ ALEJANDRO DAMIAN | 2000
(3 filas)

practicass=# CREATE SEQUENCE seq_historico
practicass=# INCREMENT BY 1
practicass=# NO MINVALUE
practicass=# NO MAXVALUE
practicass=# START WITH 1
practicass=# CACHE 1
practicass=# NO CYCLE
practicass=# OWNED BY NONE;
CREATE SEQUENCE
practicass=#
practicass=# CREATE TABLE historico (
practicass=# clave          NUMERIC(10)  DEFAULT nextval('seq_historico') NOT NULL,
practicass=# tabla         VARCHAR(20)  NOT NULL,
practicass=# accion        VARCHAR(10)  NOT NULL,
practicass=# cargo         VARCHAR(60)  NOT NULL,
practicass=# usuario       VARCHAR(60)  DEFAULT user NOT NULL,
practicass=# fecha         TIMESTAMP   DEFAULT now() NOT NULL,
practicass=# dato_viejo    VARCHAR(4000),
practicass=# dato_nuevo    VARCHAR(4000),
practicass=# CONSTRAINT pk_historico PRIMARY KEY (clave)
practicass=# );
ERROR: la relación «historico» ya existe
practicass=#
practicass=# CREATE FUNCTION f_trg_historico() RETURNS trigger AS $$
practicass=# DECLARE tcampo_new TEXT;
practicass=# DECLARE tcampo_old TEXT;
practicass=# BEGIN
practicass=# IF TG_OP = 'INSERT' THEN
practicass=#   FOR rec IN 1 .. TG_ARGV[0] LOOP

```

nuevas y originales del registro afectado, es decir, los datos que contenían antes y después de la operación que activó el trigger. PostGres nos provee de otras variables internas que podemos usar en los Trigger Procedures, y es oportuno mencionarlas en su totalidad:

TG_NAME: Contiene el nombre del trigger en ejecución

TG_WHEN: Indica si se trata de un trigger de BEFORE o AFTER.

TG_LEVEL: Su valor es "ROW" para un trigger definido para correr "FOR EACH ROW", o "STATEMENT" para los triggers que corren una única vez para toda la sentencia SQL que los disparó.

TG_OP: Contiene la sentencia SQL que provocó la ejecución del trigger. Sus valores posibles son "INSERT", "UPDATE", "DELETE" y "TRUNCATE".

TG_RELID: Almacena el oid (object ID, un valor interno del diccionario de datos de PostGres) de la tabla sobre la que se definió el trigger.

TG_RELNAME: Similar a la anterior, pero devuelve el nombre de la tabla (discontinuada).

TG_TABLE_NAME: Idem anterior, en las últimas versiones de PostGres se indica su uso en lugar de TG_RELNAME.

TG_TABLE_SCHEMA: Guarda el nombre del esquema donde reside la tabla del trigger.

TG_NARGS: Devuelve la cantidad de argumentos pasados por el trigger al trigger procedure.

TG_ARGV[]: Contiene un array (cuya longitud está especificada por la variable anterior) que almacena los valores de los argumentos pasados por el trigger al trigger procedure.

Como pueden ver, el procedimiento almacenado que escribimos para ser invocado por el trigger sólo hace uso de TG_OP, para determinar si el registro está siendo creado, modificado o borrado, y de acuerdo a ello decide si toma los valores de NEW o de OLD para volcarlos al histórico. Sólo resta escribir el trigger propiamente dicho, indicando que ante cualquier acción sobre la tabla "empleado" debe invocarse al trigger procedure que acabamos de analizar:

```

CREATE TRIGGER trg_empleado_historico
AFTER INSERT OR UPDATE OR DELETE ON empleado
FOR EACH ROW
EXECUTE PROCEDURE f_trg_empleado_historico();

```

>BASES DE DATOS-

Una vez creada la tabla, el procedimiento almacenado y el trigger, realizamos algunas operaciones sobre la tabla “empleado” para luego verificar que se hayan registrado en el histórico. Primero creamos un nuevo registro. Como en entregas anteriores preparamos la tabla para que el legajo se genere en forma automática, no es necesario especificar el campo clave:

```
INSERT INTO empleado
(cargo, apellido_y_nombre, supervisor)
VALUES
('ADMIVO', 'DOMINGUEZ ALEJANDRO', 2);
```

En nuestro caso, el valor del campo clave “legajo” que se generó para este nuevo registro es 2001. Verifiquen ustedes en su caso el valor de la clave, y tomen nota de él para aplicarlo en las dos sentencias siguientes. Ahora vamos a modificar este registro, cambiando solamente dos de los cuatro atributos que posee.

```
UPDATE empleado
SET apellido_y_nombre = 'DOMINGUEZ ALEJANDRO DAMIAN',
    supervisor = 2000
WHERE legajo = 2001;
```

Finalmente, eliminamos el registro:

```
DELETE FROM empleado
WHERE legajo = 2001;
```

Ahora, con un query SELECT consultamos la tabla de auditoría, que contendrá lo siguiente:

```
SELECT * FROM empleado_historico;
```

```
INSERT | 2011-12-12 21:22:40.171 | postgres |    2001 | ADMIVO | DOMINGUEZ ALEJAN-
DRO      |          2
```

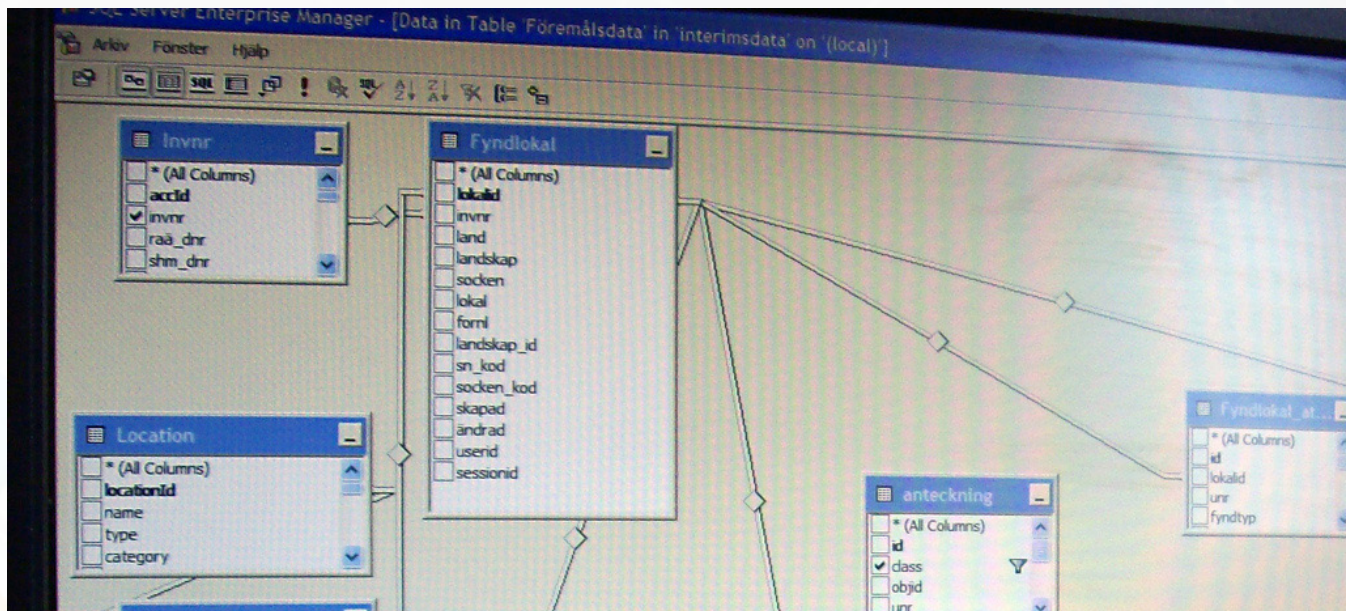
```
UPDATE | 2011-12-12 21:22:47.468 | postgres |    2001 | ADMIVO | DOMINGUEZ ALEJAN-
DRO DAMIAN |        2000
```

```
DELETE | 2011-12-12 21:22:54.281 | postgres |    2001 | ADMIVO | DOMINGUEZ ALEJAN-
DRO DAMIAN |        2000
```

```
(3 filas)
```

SEGUNDA SOLUCIÓN: ESCALABLE Y ÓPTIMA

Este enfoque tiene un grado mayor de complejidad que el anterior, pero posee unas virtudes interesantes. Probablemente la más notoria es que puede guardarse el registro de todas las tablas que se quieran auditar en una única tabla, que llamaremos “historico”, y para completarla vamos a escribir un solo trigger procedure. Además, cuando se actualice un registro (UPDATE), en esta tabla histórica sólo se guardarán aquellos campos que han sido modificados, lo cual implica un ahorro de espacio de storage considerable, sobre todo para una base de datos que sufre un gran volumen de operaciones diarias. Lo que no puede evitarse -igual que en la primera variante- es la creación de un trigger para cada tabla a auditar. El punto débil de esta versión del registro de auditoría es que la estructura genérica de la tabla histórica provoca que reconstruir un registro a un estado previo determinado y presentarlo con la misma estructura que poseía en su forma original implicará escribir un programa bastante arduo y complejo, merecedor de



una nota en sí mismo. Naturalmente, siempre puede leerse el histórico en forma plana, y rearmar “manualmente” cada registro y los estados por los que ha pasado.

```
CREATE SEQUENCE seq_historico
  INCREMENT BY 1
  NO MINVALUE
  NO MAXVALUE
  START WITH 1
  CACHE 1
  NO CYCLE
  OWNED BY NONE;
```

```
CREATE TABLE historico (
  clave          NUMERIC(10)   DEFAULT nextval('seq_historico') NOT NULL,
  tabla          VARCHAR(63)   NOT NULL,
  accion         VARCHAR(10)   NOT NULL,
  campo          VARCHAR(63)   NOT NULL,
  usuario        VARCHAR(63)   DEFAULT user NOT NULL,
  fecha          TIMESTAMP     DEFAULT now() NOT NULL,
  dato_viejo     VARCHAR(4000),
  dato_nuevo     VARCHAR(4000),
  CONSTRAINT pk_historico PRIMARY KEY (clave)
);
```

Dada la estructura genérica de la tabla histórica, necesitamos una clave artificial para poder identificar unívocamente cada registro que almacenemos en ella. Para generar la clave recurrimos a una secuencia, como ya hemos visto en entregas anteriores. Al crear este histórico estamos asumiendo algunas características del modelo de datos que, de variar, provocarán que el esquema de auditoría no funcione. En primer lugar, suponemos que no se han definido

>BASES DE DATOS-

“En el caso de un UPDATE, podemos comparar el OLD y el NEW de cada atributo, para grabar en el histórico solamente los que han cambiado.”

“También es interesante apreciar cómo una función tan vital en algunos casos como los registros de auditoría pueden encararse de maneras muy distintas, una sencilla de escribir pero que requiere más trabajo cada vez que agreguemos una nueva tabla al listado de entidades a monitorear, y otra más compleja, pero de características “WORM” (“Write Once, Reuse Many”).”

identificadores (nombres de tablas, campos y usuarios) de longitud mayor que la soportada por defecto por PostGres (64 -1). Segundo, y todavía más importante, la precisión de ningún campo del modelo de datos debe superar los 4000 caracteres para poder almacenar sus valores en el registro histórico. Como en PostGres los campos de tipo VARCHAR no tienen una longitud máxima fija, como sucede en otros motores, sino que pueden contener hasta aproximadamente 1 Gb de texto, esta consideración es especialmente importante, y debe ajustarse en forma acorde a la mayor precisión especificada en cualquier atributo perteneciente a una tabla a auditar.

```
CREATE FUNCTION f_trg_historico() RETURNS trigger AS
$$
    DECLARE tcampo_new TEXT;
    DECLARE tcampo_old TEXT;
    BEGIN
        IF TG_OP = 'INSERT' THEN
            FOR rec IN 1 .. TG_ARGV[0] LOOP
                EXECUTE 'SELECT CAST ( $1 . ' || TG_
                ARGV[rec] || ' AS TEXT)' INTO tcampo_new USING NEW;
                INSERT INTO historico
                (tabla, accion, campo, dato_nuevo)
                VALUES
                (TG_TABLE_NAME, TG_OP, TG_ARGV[rec], tcam-
                po_new);
            END LOOP;
            RETURN NEW;
        END IF;

        IF TG_OP = 'DELETE' THEN
            FOR rec IN 1 .. TG_ARGV[0] LOOP
                EXECUTE 'SELECT CAST ( $1 . ' || TG_
                ARGV[rec] || ' AS TEXT)' INTO tcampo_old USING OLD;
                INSERT INTO historico
                (tabla, accion, campo, dato_nuevo)
                VALUES
                (TG_TABLE_NAME, TG_OP, TG_ARGV[rec], tcam-
                po_old);
            END LOOP;
            RETURN OLD;
        END IF;
```



```

IF TG_OP = 'UPDATE' THEN
  FOR rec IN 1 .. TG_ARGV[0] LOOP

    EXECUTE 'SELECT CAST ( $1 . ' || TG_ARGV[rec] || ' AS TEXT)' INTO
tcampo_old USING OLD;
    EXECUTE 'SELECT CAST ( $1 . ' || TG_ARGV[rec] || ' AS TEXT)' INTO
tcampo_new USING NEW;

    IF tcampo_new <> tcampo_old THEN
      INSERT INTO historico
      (tabla, accion, campo, dato_viejo, dato_nuevo)
      VALUES
      (TG_TABLE_NAME, TG_OP, TG_ARGV[rec], tcampo_old, tcampo_new);
    END IF;
  END LOOP;

  RETURN NEW;
END IF;

END;
$$ LANGUAGE plpgsql;

```

El trigger procedure debe recibir como argumento todos los campos del registro afectado por el trigger. Al igual que en la implementación sencilla, se vale de TG_OP para determinar si el registro está siendo creado, eliminado o modificado. Luego, recurre a un iterador (FOR..LOOP) y a una “triquiñuela” de SQL dinámico (mediante el empleo de la sentencia EXECUTE) para generar en el histórico un registro para cada campo afectado por la operación que disparó el trigger.

En el caso de un INSERT o DELETE, naturalmente los afectados son todos los atributos de la tabla auditada, de los cuales tenemos una copia en la variable interna NEW y OLD, respectivamente. En el caso de un UPDATE, podemos comparar el OLD y el NEW de cada atributo, para grabar en el histórico solamente los que han cambiado. Es importante destacar que si alguno de los campos auditados puede contener valores nulos, esta comparación puede fallar, ya que el uso de NULL implica considerar una lógica ternaria (3VL). Si se da el caso, podemos valernos de algún truco para evitar este problema, como el empleo de la función COALESCE en conjunto con alguna constante arbitraria.

id	nombre	tabla	accion	campo	usuario	fecha	dato_viejo	dato_nuevo
1	1	empleados	INSERT	leanta	postgres	2011-12-12 21:55		2803
2	2	empleados	INSERT	carlos	postgres	2011-12-12 21:55		ADRIVO
3	3	empleados	INSERT	adelgado y nandez	postgres	2011-12-12 21:55		MULLER ERNESTO
4	4	empleados	INSERT	superwiser	postgres	2011-12-12 21:55		2
5	5	empleados	UPDATE	adelgado y nandez	postgres	2011-12-12 21:55		MULLER ERNESTO
6	6	empleados	UPDATE	superwiser	postgres	2011-12-12 21:55	2	2800
7	7	empleados	DELETE	leanta	postgres	2011-12-12 21:55		2803
8	8	empleados	DELETE	carlos	postgres	2011-12-12 21:55		ADRIVO
9	9	empleados	DELETE	adelgado y nandez	postgres	2011-12-12 21:55		MULLER ERNESTO
10	10	empleados	DELETE	superwiser	postgres	2011-12-12 21:55		2800

>BASES DE DATOS-

```
CREATE TRIGGER trg_empleado_historico2
AFTER INSERT OR UPDATE OR DELETE ON empleado
FOR EACH ROW
EXECUTE PROCEDURE f_trg_historico(4, legajo, cargo, apellido_y_nombre, supervisor);
```

Sólo resta crear el trigger propiamente dicho. A diferencia de la otra implementación, desde aquí debemos enviar como argumentos al trigger procedure todos los atributos de la tabla a la que asociamos el trigger. Noten que el primer argumento es un valor numérico, indicando la cantidad de atributos que vamos a enviar como parámetros a continuación.

Igual que antes, generamos un nuevo registro, lo modificamos y borramos, para comprobar que todo funcione de acuerdo a lo pensado:

```
INSERT INTO empleado
(cargo, apellido_y_nombre, supervisor)
VALUES
('ADMIVO', 'MULLER ERNESTO', 2);
```

```
UPDATE empleado
SET apellido_y_nombre = 'MULLER ERNST',
    supervisor = 2000
WHERE legajo = 2002;
```

```
DELETE FROM empleado
WHERE legajo = 2002;
```

Nuevamente corremos un query SELECT para verificar el registro de auditoría, con el siguiente resultado. Aunque las operaciones INSERT y DELETE generan tantos registros en esta tabla como atributos tenga la tupla implicada, en el caso de un UPDATE se graba sólo una fila para cada atributo que se haya modificado.



```
SELECT * FROM historico;
```

```

1 | empleado | INSERT | legajo | postgres | 2011-12-12
21:59:18.281 | | 2002
2 | empleado | INSERT | cargo | postgres | 2011-12-12
21:59:18.281 | | ADMIVO
3 | empleado | INSERT | apellido_y_nombre | postgres | 2011-12-12
21:59:18.281 | | MULLER ERNESTO
4 | empleado | INSERT | supervisor | postgres | 2011-12-12
21:59:18.281 | | 2
5 | empleado | UPDATE | apellido_y_nombre | postgres | 2011-12-12 21:59:45
| MULLER ERNESTO | MULLER ERNST
6 | empleado | UPDATE | supervisor | postgres | 2011-12-12 21:59:45
| 2 | 2000
7 | empleado | DELETE | legajo | postgres | 2011-12-12
21:59:54.781 | | 2002
8 | empleado | DELETE | cargo | postgres | 2011-12-12
21:59:54.781 | | ADMIVO
9 | empleado | DELETE | apellido_y_nombre | postgres | 2011-12-12
21:59:54.781 | | MULLER ERNST
10 | empleado | DELETE | supervisor | postgres | 2011-12-12
21:59:54.781 | | 2000

```

```
(10 filas)
```

EN RESUMEN

Con lo expuesto, nuevamente ratificamos el poder y la flexibilidad que aportan los triggers de base de datos en general y su implementación mediante el lenguaje PL/pgSQL en particular. También es interesante apreciar cómo una función tan vital en algunos casos como los registros de auditoría pueden encararse de maneras muy distintas, una sencilla de escribir pero que requiere más trabajo cada vez que agreguemos una nueva tabla al listado de entidades a monitorear, y otra más compleja, pero de características “WORM” (“Write Once, Reuse Many”), tal como se conoce a este tipo de programas en el ámbito académico de los países angloparlantes. Según acostumbramos, concluimos la presente entrega indicando la referencia bibliográfica para quienes deseen ahondar en esta temática. El manual online de PostGres es muy completo, y pueden encontrar lo necesario como para convertirse en expertos escritores de triggers en las páginas [<http://www.postgresql.org/docs/9.0/static/sql-createtrigger.html>] y [<http://www.postgresql.org/docs/9.0/static/plpgsql-trigger.html>]. Nos despedimos hasta el próximo mes, no sin antes invitarlos a escribirnos para transmitirnos sus comentarios, inquietudes, sugerencias, o cualquier otra cosa pertinente a esta serie de notas sobre un área de las tecnologías de la información tan fascinante e importante como lo es el mundo de las bases de datos relacionales. ¡Hasta la próxima! **P**



Novedades en Android para Developers

Por Mariano Salvetti

mariano.salvetti@pixelscode.com

@dudoSalvetti



En octubre del 2011, se publicó el SDK de Android 4.0 donde pudimos ver por encima las novedades que traía esta versión del Sistema Operativo Android, como una nueva versión dedicada a las Tablets y a los SmartPhones. En este artículo, primero veremos las herramientas y novedades que ofrece la versión de Android 3.0 a los desarrolladores, presentada en Julio de 2011, para así luego pasar a la última versión Android 4.0, siempre con una mirada de Developer.

NOVEDADES EN ANDROID 3.0 SDK

La principal novedad de Android 3.0 (HoneyComb) es la interfaz exclusiva para Tablets, que aprovecha todas las posibilidades de las pantallas más grandes, donde gracias a la nueva barra de sistema y la de acciones, se hacen mucho más intuitivas las aplicaciones y el sistema en general.

NUEVAS HERRAMIENTAS EN LA INTERFAZ DE USUARIO

Ahora los desarrolladores pueden dividir las aplicaciones en módulos, llamados fragmentos, para que una aplicación se divida en subcomponentes y poder manejarlos de manera individual y combinarlos de distintas formas como, por ejemplo, la creación de un

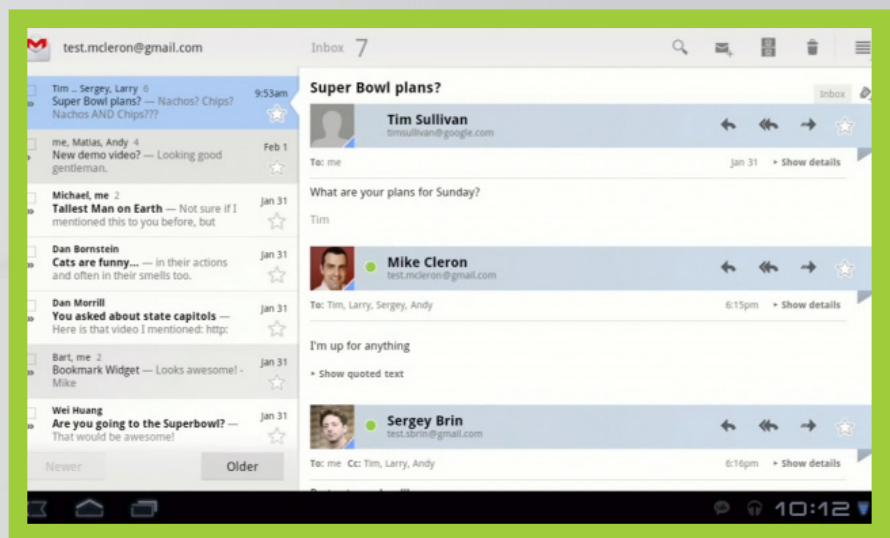
multipanel, como se muestra en la imagen a continuación.

Una aplicación podrá estar formada por varios fragmentos, donde cada uno es una vista de la aplicación para interactuar de forma independiente. Se pueden agregar, eliminar y mover dentro de una aplicación. Al ser modulares se facilitará la adaptación de las aplicaciones a todo tipo de pantallas. ¿Por qué fragmentos? Desde el punto de vista de un desarrollador, la transición más grande de SmartPhones a Tablets es el tamaño de la pantalla. Entonces, los fragmentos crean un puente para las aplicaciones que desean ser optimizadas. Más información en: <http://goo.gl/luYel>

También se ha actualizado la interfaz de los widgets para aprovechar las pantallas de las Tablets usando la nueva interfaz holográfica. Se aumentan los tipos de widgets, como las nuevas pilas en 3D, listas, cajas de búsqueda, calendarios, etc. Los developers pueden crear sus widgets heredando el nuevo diseño. Entonces con esto, los widgets son más interactivos, ya que se pueden crear colecciones mediante pilas 3D, listas o tablas para que los usuarios puedan ver su contenido de forma rápida y sencilla.

También ahora cada aplicación podrá tener su barra de acción, que está ubicada en la parte superior. Las acciones de esta barra podrán depender del contexto de lo que estemos realizando. Y por otro lado, se presentaron mejoras en las notificaciones, ya que se han mejorado las notificaciones para la interfaz de las Tablets. Se podrán crear notificaciones mostrando más contenido y mejor ordenado, ya que se podrán usar imágenes más grandes, un título, decir el grado de prioridad de esa notificación y el resto de propiedades en versiones anteriores de Android. A continuación tenemos una imagen del nuevo escritorio:

Android 3.0 da soporte a la aceleración por hardware de OpenGL, y



**“ Hemos en-
contrado un
pequeño problema de “seguridad” a mi criterio: con una
foto también podemos des-
bloquear el teléfono, ya que
detecta la cara en la foto.”**

así se tendrá todo el control para hacer uso de esta aceleración por hardware. También se han presentado nuevas herramientas para crear animaciones, es decir, será más fácil realizar animaciones en las aplicaciones, como crear transiciones, cambios de color o desvanecimien-

tos, entre otros efectos. Por último, en relación a las mejoras en la parte gráfica, debemos destacar que se incluye el motor gráfico RenderScript 3D que es un motor gráfico con su propia API para la creación de gráficos en 3D. Con él se facilitará la creación de contenido 3D, como efectos, fondos de pantalla, carruseles y mucho más. Un ejemplo es la aplicación de Google para libros que se muestra a continuación:

¿Y LA COMPATIBILIDAD?

Android 3.0 trae una nueva interfaz diseñada para las Tablets, pero las actuales aplicaciones para versiones anteriores a SDK 3.0 no tendrán problemas en funcionar en esta nueva versión. Tan sólo tenemos que agregar un nuevo atributo a sus aplicaciones para que estas funcionen sin problemas en

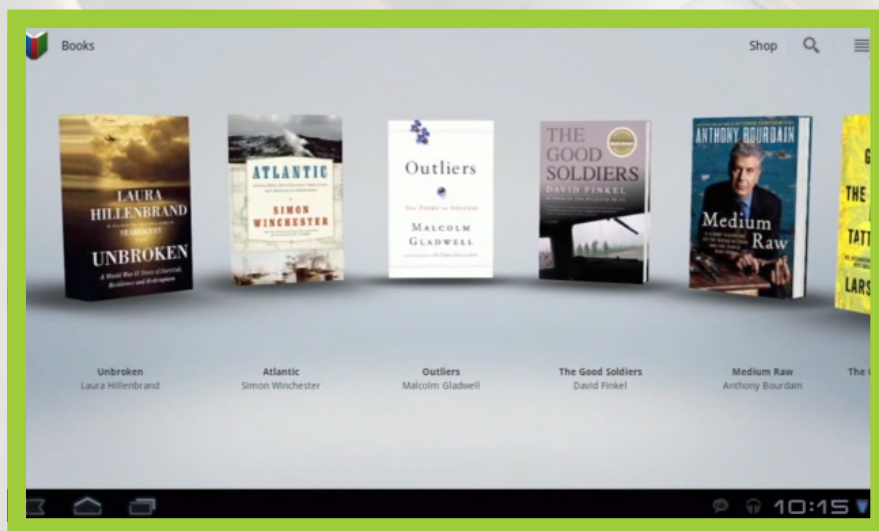
>DESARROLLO PARA MÓVILES-



la interfaz para Tablets. Ese nuevo atributo lo que hará es añadir la barra de acción, donde veremos la tecla Menú, ya que la mayoría de las Tablets que salen con Android 3.0 no tienen botones físicos ni táctiles.

NOVEDADES EN ANDROID 4.0 SDK

Retrasos de Google, lutos de por medio, al final, el 19 de Octubre del 2011, Google presentó en sociedad la versión Android 4.0 llamada Ice Cream Sandwich, que ve la luz con mucho amor para los desarrolladores. Repasemos a continuación las mejoras y novedades que nos trae, destacando que esta versión unifica el uso en cualquier dispositivo, es decir, podemos desarrollar aplicaciones que se ejecuten en teléfonos, tablets, televisores y netbooks, utilizando la nueva interfaz limpia y moderna con una nueva fuente llamada "Roboto", muy al estilo de Honeycomb.



UNIFICANDO EL CAMINO

La versión de sistema operativo Google Android 4, conocida como Ice Cream Sandwich, ha sido presentada junto con el Smartphone Galaxy Nexus de Samsung. Afortunadamente, Google sigue los pasos dados por Apple, unificando el sistema de ambas modalidades de productos, con el fin de ayudar a los desarrolladores. Como resultado, tanto las tabletas de Android,

como los teléfonos, van a compartir la misma interfaz de usuario. Y esto significa poseer nuevas herramientas de la interfaz, mejores prácticas de diseño, código simplificado y más recursos para el desarrollo.

Esta versión unificada para móviles y tablets, logra que por fin Fragments, Loaders, la ActionBar y otras mejoras de Honeycomb estén disponibles para Smartphones de forma nativa.

También se presentaron funciones nuevas para la cámara. Exposición ZSL, enfoque continuo y zoom de imagen, funciones que permiten a las aplicaciones capturar mejores imágenes fijas y de vídeo. Las aplicaciones tienen la posibilidad de capturar instantáneas de alta resolución durante la grabación de vídeo. Las aplicaciones pueden establecer regiones personalizadas de medición en una previa de la cámara y después manejar el balance de blancos y la exposición dinámicamente para esas regiones. Para facilitar el enfoque y tratamiento de imágenes, un servicio de detección de rostros identifica y compara las caras en una vista previa y devuelve sus coordenadas en la pantalla.

MEJORAS EN COMUNICACIONES Y DE CONECTIVIDAD

Android 4.0 extiende las características sociales y la distribución de cualquier aplicación en el dispositivo. Las aplicaciones pueden integrar los contactos, los datos del perfil, y eventos de calendario a partir de cualquiera de las actividades de los usuarios o redes sociales.

Se presentaron nuevas APIs sociales, y ahora tenemos la oportunidad de sumar nuevas características orientadas a las redes sociales a nuestro perfil en el teléfono, y como desarrolladores, podemos explotar este hecho a través del `ContactsContract`. Pero, además, podremos ofrecerle al usuario la posibilidad de añadir contactos desde nuestras aplicaciones.

También, existe una API nativa para acceder a Google Calendar, llamada `Calendar Provider`. Este proveedor de calendario y de contenido compartido en la API nos va a facilitar a los desarrolladores agregar servicios de calendario a las aplicaciones. Con el permiso del usuario, cualquier aplicación puede agregar eventos a la base de datos compartida y gestión de las fechas, los asistentes, las alertas y recordatorios. Las solicitudes también se pueden leer, las entradas de la base de datos, incluyendo los eventos aportados por otras aplicaciones y controlar la presentación de alertas de eventos y recordatorios. Mediante el proveedor de calendario, las aplicaciones pueden tomar ventaja de los datos de eventos provenientes de una variedad de aplicaciones y protocolos,

para ofrecer formas innovadoras de visualización y gestión de eventos de un usuario. Las aplicaciones también pueden usar datos de la agenda para mejorar su contenido.

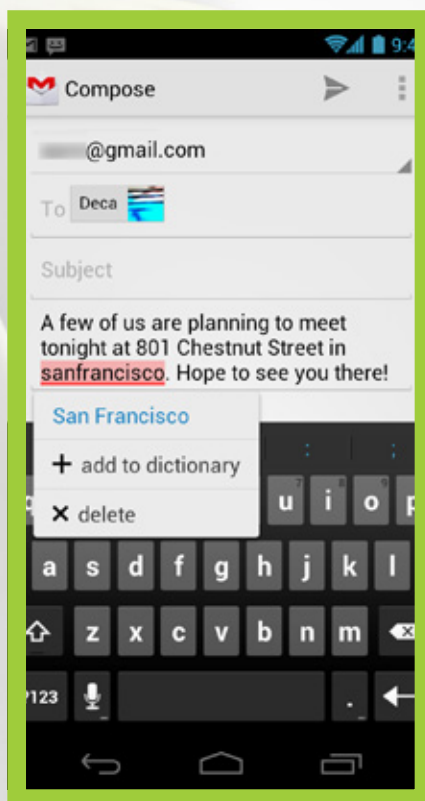
Por medio del `Media Effects Framework` tenemos disponible un conjunto de utilidades que permite agregar efectos visuales a imágenes y vídeos.

Por otro lado, una mejora interesante es la de la detección de caras: si bien ya existía el `FaceDetection` en Android desde su versión 1.0, librerías como `OpenCV` obtenían mejores resultados y rendimiento. Con las nuevas mejoras introducidas, y sobretodo pudiendo utilizar esta API sin recurrir al `NDK`, ¿ganará el `SDK`? Hemos encontrado un pequeño problema de “seguridad”





El IME es un método de entrada que implementa un modelo particular de interacción, que permite al usuario generar texto. El sistema se une al método de entrada actual que está en uso, permitiendo que se pueda crear y ejecutar, y le dice cuándo se tiene que ocultar y mostrar su interfaz de usuario.”



a mi criterio: con una foto también podemos desbloquear el teléfono, ya que detecta la cara en la foto. ¿Android Beam? En mi opinión, es la novedad más importante del SDK. Recordemos que NFC son las siglas en inglés de Near Field Communication, una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos a menos de 10cm. Ya no sólo podemos utilizar NFC, sino que ahora tenemos una API dedicada a la transferencia de información entre terminales vía NFC. Y lo mejor es que el sistema funciona

tipo Push, por lo que presumiblemente el receptor del mensaje no tendrá que iniciar la aplicación de destino para recibirlo, sino que Android Beam lanzará un Intent cuando esté el mensaje disponible. Más información en: <http://goo.gl/ZtmiQ>

Por distintos medios en internet sabemos que existen muchos aparatos médicos que utilizan Bluetooth para enviar y recibir información. Google nos ofrece una herramienta para facilitar el desarrollo en un campo que no es el del típico usuario, pero que tiene enormes posibilidades, mediante Bluetooth Health Device Profile.

También aparece una API de Servicios de Textos para integrar Correctores Ortográficos, es decir, Android 4.0 permite a las aplicaciones consultar los servicios de texto disponibles, tales como diccionarios y correctores ortográficos para sugerir palabras, correcciones y datos similares. Los servicios de texto son externos al IME activo, por tanto los desarrolladores pueden crear y distribuir diccionarios y motores de sugerencia que se conectan a la plataforma. Cuando una aplicación recibe resultados de un servicio de texto, por ejemplo, sugerencias de palabras, puede mostrarlas en una



ventana emergente directamente dentro de la vista de texto, en lugar de confiar en el IME para mostrarlas. Pero ¿qué es el IME? El IME es un método de entrada que implementa un modelo particular de interacción, que permite al usuario generar texto. El sistema se une al método de entrada actual que está en uso, permitiendo que se pueda crear y ejecutar, y le dice cuándo se tiene que ocultar y mostrar su interfaz de usuario. Sólo un IME funciona a la vez.

CONCLUSIONES

Gracias a Android 3.0, los desarrolladores podemos ofrecer aplicaciones más intuitivas y vistosas con una nueva experiencia que nos hará el día a día más fácil como desarrolladores, sumado a que tenemos un mejor soporte para re-

des WIFI y para una gran variedad de periféricos con conexión USB, como teclado, Mouse y cámaras digitales, y también con la nueva interfaz visual y a la posibilidad de utilizar los Fragmentos, tenemos herramientas más que interesantes para desarrollar aplicaciones muy intuitivas. Android 4.0 amplía las funciones sociales y para compartir. Google quiere potenciar el elemento social, gracias a la API SOCIAL y API CALENDAR, entre las mejoras en comunicaciones que también presentaron.

Está claro que Android es una de las plataformas móviles con mayor popularidad en el mercado. Si algo tenemos que destacar de estas dos últimas versiones, es que los desarrolladores tenemos todas las facilidades para adaptar las apli-

caciones a la nueva interfaz y así mejorar la experiencia y usabilidad de las mismas, con interesantes APIs, mejoras en las capacidades multimedia y en los nuevos tipos de conectividad.

Desde antes de su lanzamiento, Android 4.0 ha sido la esperanza de los desarrolladores con el fin de que de una vez por todas se unificaran ambas plataformas de dispositivos. A mi criterio, queda mucho por mejorar y que de una buena vez se unifiquen Google TV, con Tablet y Smartphone, ya en Android 5.0, que se podría llamar "Jelly Bean".

Pero, paso a paso se está llegando. ¡Hasta la próxima entrega! **P**

Google Dart

El nuevo JavaScript

Cada vez resulta más evidente que la guerra entre las empresas de software o que se dedican a este rubro de una u otra forma, han decidido mover el campo de batalla a Internet y esas cuasi computadoras de bolsillo que son nuestros celulares. Para el primer plano de ataque, el de la Web, muchos frentes se están abriendo y el apurarse para captar la atención de los desarrolladores parece ser una de las tácticas para ganar esta batalla. De esta forma Google ha decidido intentar otra avanzada al presentar un nuevo lenguaje de programación tratando de suplantar uno de los que nació prácticamente con Internet.



Por Matías Iacono

matias.iacono@pixelscode.com

GANANDO ADEPTOS

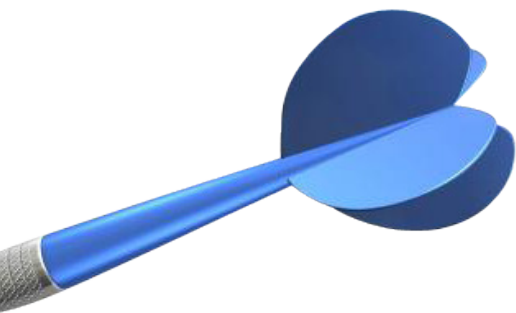
En la historia de Google podemos encontrar fácilmente muchos proyectos con los cuales ha intentado ganar seguidores, aunque algunos con mejores resultados que otros, pero todos dirigidos a aumentar la cantidad de personas que usen sus servicios. Seguramente, se nos vendrán a la cabeza diferentes proyectos como Wave, Buzz, Orkut, entre otros. Todos estos prometían una serie de soluciones muy interesantes de cara al usuario de aplicacio-

nes, pero por diferentes motivos no llegaron a conseguir la masa crítica necesaria para dejar una marca rentable por lo que fueron dados de baja por la empresa.

En todo caso, las aplicaciones anteriores tenían como público objetivo al usuario de aplicaciones, pero la misma historia se ha repetido hacia otro público, el de los desarrolladores. Posiblemente algunos pocos habrán escuchado hablar de Go, un lenguaje que intentó de alguna forma

tratar de captar la atención de la comunidad de desarrolladores con un posible no tan rotundo éxito como se lo esperaba.

Actualmente Google, y con la llegada, definitiva, cada vez más cercana de HTML5, volvió al ruedo con esta idea de crear un nuevo lenguaje de programación que traiga nuevos adeptos hacia sus filas. Google Dart, entonces, pretende redefinir, por una parte, la forma en cómo podemos crear código



para interactuar, por ejemplo, con el dominio de HTML y HTML5 con cualidades especiales, suplantando el uso de JavaScript tradicional por una implementación que se asemeja al lenguaje Java, y por otra, crear aplicaciones en el servidor para la creación de contenido dinámico, sean sitios Web u otro tipo de aplicaciones.

De esta forma, podemos entender que un lenguaje con mayor estructura, con la posibilidad de hacer uso de tipos de datos y que, además, soporte el dinamismo de

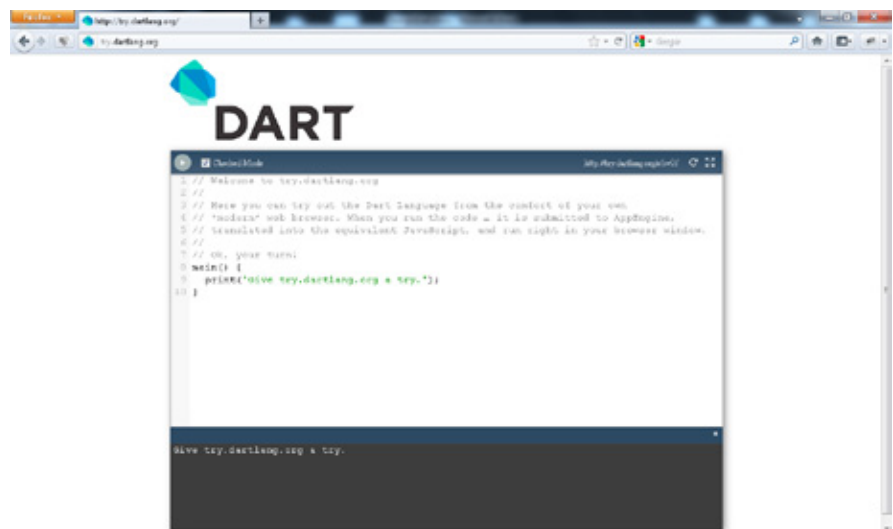
y, por lo tanto, vean a este lenguaje como una buena oportunidad para moverse al mundo del desarrollo Web dinámico.

¿Y CUÁL ES EL PROBLEMA CON JAVASCRIPT?

Probablemente, nos preguntemos del porqué de un nuevo lenguaje de programación y qué tiene de malo el existente. En este punto, las respuestas o justificaciones que se pueden recoger en foros oficiales de Dart no son muy alentadoras, ya que no existe una justificación directa o clara para dejar de lado el uso de

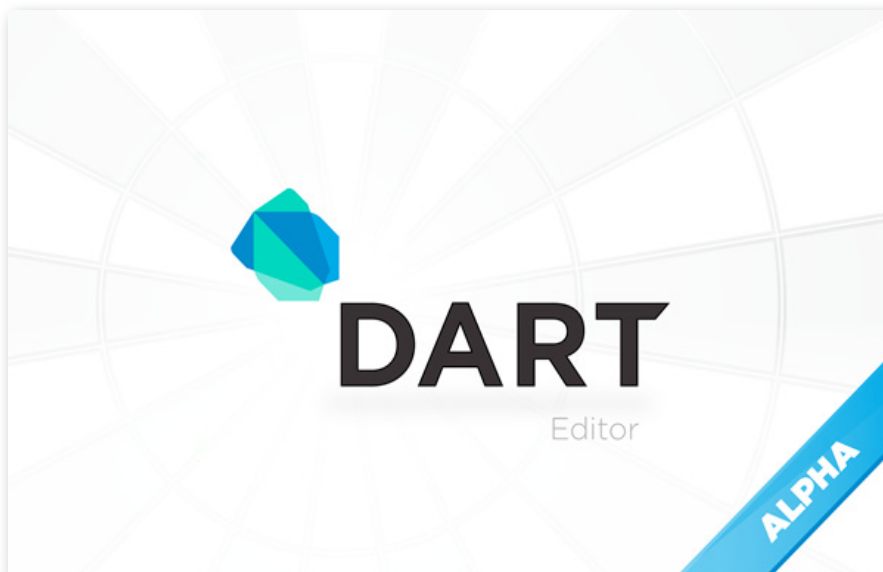


Un lenguaje con mayor estructura, con la posibilidad de hacer uso de tipos de datos y que, además, soporte el dinamismo de JavaScript, en el caso de la Web, puede hacer que aquellos desarrolladores acostumbrados a utilizar lenguajes como Java o C# se sientan como en casa con Google Dart.”



JavaScript, en el caso de la Web, puede hacer que aquellos desarrolladores acostumbrados a utilizar lenguajes como Java o C# se sientan como en casa con Google Dart

Un editor de código fuente para Google Dart que compilará nuestro código y mostrará los resultados en el navegador.

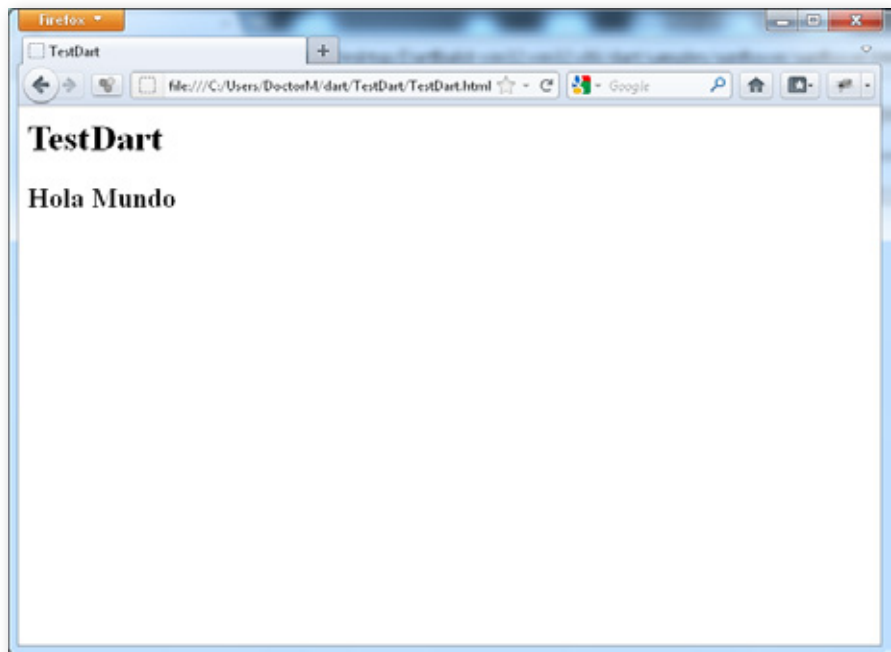


.....
*Dart Editor desplegándose para
iniciar el trabajo.*
.....

JavaScript (en el caso de la Web y ambientes que lo soporten), ya que normalmente se decantan principalmente por las capacidades de herencia, polimorfismo y tipos fuertes de datos, así como la elegancia en la escritura. Por el mismo motivo, defensores de JavaScript argumentan como inválidos estos argumentos o poco fuertes para necesitar un nuevo lenguaje de programación ya que, a pesar de presentar algunas peculiaridades, muchas de estas características están presentes en JavaScript.

LAS HERRAMIENTAS DE DESARROLLO

A pesar de la duda que se nos pudiera presentar sobre el uso o no de Google Dart, el futuro del mismo,



sus proyecciones y posibilidades, siempre será interesante acercarse un poco para husmear y estar preparados por si el mercado se decantara hacia el uso del mismo, y para poder hacer esto será necesario primero tener las herramientas adecuadas a mano.

.....
*Al ejecutar el "Hola Mundo", éste es
mostrado correctamente en Firefox.*
.....

Si no queremos descargar o instalar ningún tipo de componente, podremos realizar algunas pruebas rudimentarias del lenguaje desde la Web <http://try.dartlang.org/>. En esta página podremos crear código de prueba de Google Dart y ejecutarlo, visualizando los resultados del mismo al pie de la página.

Si, por el contrario, necesitáramos hacer algo de código más robusto, utilizando librerías más específicas, podremos utilizar una versión en estado beta de Dart Editor basada en Eclipse. Esta herramienta nos permitirá escribir código separado en diferentes archivos, importar diferentes librerías de Dart, y compilar y probar el resultado en diferentes navegadores Web.

Para descargar la última versión de Dart Editor podremos hacerlo desde la Web oficial del lenguaje Dart (<http://www.dartlang.org/>). Al momento de escribir este artículo, la última versión es la 1910.

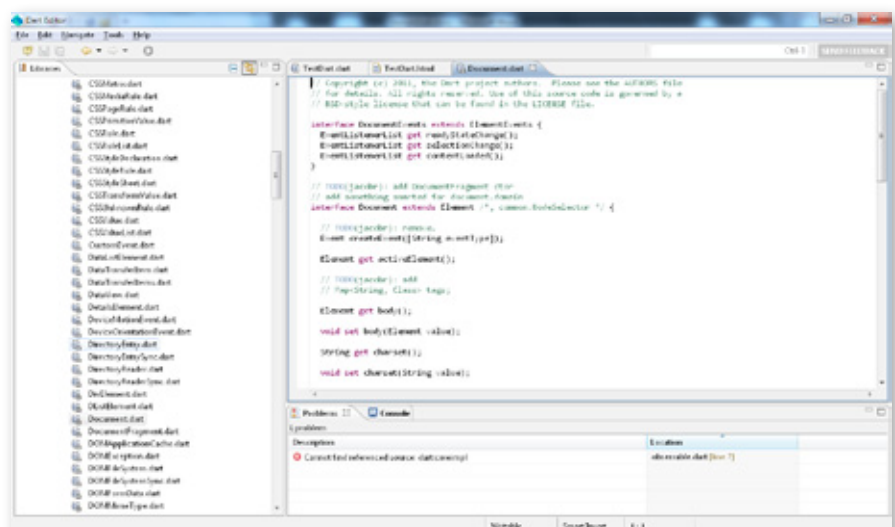
PRIMERO PASOS EN GOOGLE DART

En el momento que creamos un nuevo proyecto de Google Dart notaremos que además de la primera

clase con código, también contamos con una página HTML por defecto. Esta página Web será la encargada de ejecutar el código que escribamos.

Tenemos que tener en cuenta que Dart aún está en etapas de producción, por lo que no existe soporte directo por parte de los distintos navegadores para este lenguaje. Una de las ideas rondando la implementación definitiva es que los navegadores incluyan una máquina virtual de ejecución que soporte Dart. Otra posibilidad, que es el camino a tomar en la actualidad, es que al compilar nuestro código, éste sea transformado en JavaScript para que los diferentes navegadores puedan trabajar con él, aunque aquí deberemos tener cuidado, ya que no todos los navegadores soportan el resultado final de compilación (Firefox y Chrome

como los principales) y, por otro lado, que la firma del framework llega a ocupar 6 o 10 megabytes, algo inviable si queremos publicar nuestro código en la Web, aunque podremos optar por una compilación optimizada, la que reducirá el tamaño de 6 megabytes a, dependiendo de nuestra aplicación, unos 200 kilobytes.



Desde Dart Editor es posible inspeccionar la lista de todos los objetos importados y así aprender su utilización.



“Google Dart puede ser el siguiente lenguaje de desarrollo y con el cual Google pueda apuntalarse como proveedor de tecnología de desarrollo sólida, y no sólo como proveedor de servicios para usuarios finales.”

```
<html>
  <head>
    <title>TestDart</title>
  </head>
  <body>
    <h1>TestDart</h1>
    <h2 id="status">dart
is not running</h2>
    <script type="text/
javascript" src="TestDart.
dart.app.js"></script>
  </body>
</html>
```

Al inspeccionar el código HTML de la página incluida en nuestro proyecto, vemos que no contiene ningún elemento extraño adicional, mas sí una referencia al código JavaScript que será generado por nuestro proyecto. Siendo que el código trabajará dentro de este contexto, entonces podremos tomar los diferentes elementos HTML y manipularlos desde Dart.

```
#import('dart:html');

class TestDart {
  TestDart() {
  }
  void run() {
    write("Hola Mundo");
  }
  void write(String message) {
    document.
query('#status').innerHTML
= message;
  }
}
```

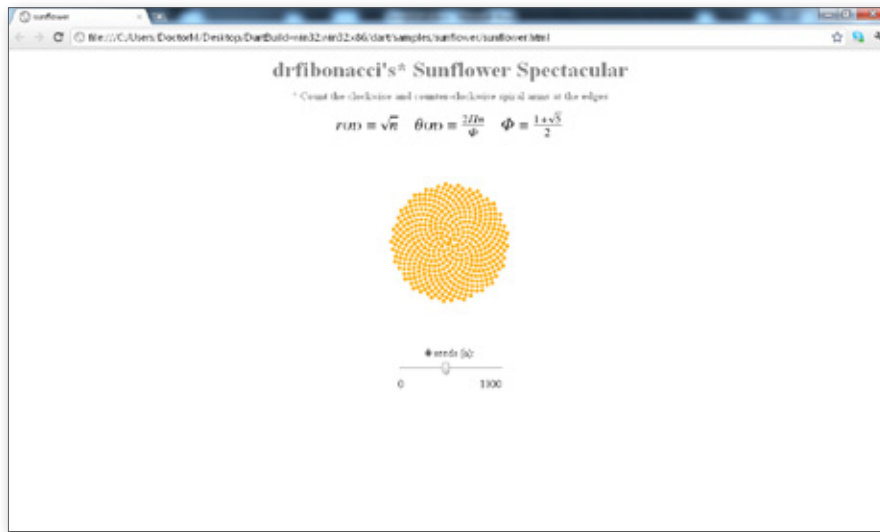
```
}
void main() {
  new TestDart().run();
}
```

El código de Dart presentado aquí es el ejemplo clásico del “Hola Mundo”, pero nos sirve para ilustrar algunas de las cualidades del lenguaje. Podemos ver que la estructura es similar a lenguajes orientados a objeto, pudiendo declarar clases mediante el uso de la directiva class, con una función main() como punto de entrada de nuestro programa o el uso de void para representar funciones sin retornos de tipos.

Como todo lenguaje basado en librerías de código, las piezas de código que importemos desde el framework de trabajo nos darán la posibilidad de interactuar con el entorno en el que estemos desarrollando. En nuestro código, la importación de #import('dart:html'); es la que nos permite interactuar con el código HTML de forma tradicional, esto es, pudiendo consultar los objetos contenidos en la página y modificar sus atributos.

GOOGLE DART Y HTML5

Como comentábamos al principio, uno de los puntos interesantes que presenta Dart es la posibilidad de interactuar desde el lenguaje con la nueva especificación de HTML5. Podremos encontrar dentro del lenguaje una completa implementación y soporte para todos los



La manipulación del tag canvas de HTML5 es transparente para el desarrollador desde Google Dart.

elementos propuestos en HTML5, desde el uso de canvas, audio y video, hasta geolocalización y otros.

El proyecto Sunflower (Girasol), incluido como ejemplo dentro de Dart Editor, puede ser uno de los mejores puntos de partida para aprender más sobre cómo interactuar con el nuevo estándar. En la página Web de ejecución del código podremos encontrar la definición de un objeto canvas.

```
...
<div>
  <canvas
    id="canvas" width="300"
    height="300"></canvas>
</div>
...
```

Este lienzo será el utilizado por Dart para dibujar el resultado de la interacción del código con el usuario.

```
var doc = window.document;
HTMLCanvasElement canvas = doc.
  getElementById("canvas");
ctx = canvas.
  getContext("2d");
...
```

Como vemos, la implementación getContext del objeto canvas y de éste como objeto propio de Dart se encuentran dentro del lenguaje. El tipo HTMLCanvasElement como una envoltura para tag HTML canvas contiene la funcionalidad para interactuar con el mismo modelo en HTML pero desde el lenguaje.

De la misma forma, los demás atributos y objetos presentes en JavaScript para la manipulación de HTML5 también tendrán su representación en Dart.

RESUMEN

Google Dart puede ser el siguiente lenguaje de desarrollo y con el cual

Google pueda apuntalarse como proveedor de tecnología de desarrollo sólida, y no sólo como proveedor de servicios para usuarios finales. De cualquier manera, uno de los sustentos que motiva la creación del mismo remplazando el actual JavaScript y toda la implementación tecnológica ya existente, puede ser un elemento que juegue en contra de su aceptación. Por otra parte, aún faltará un largo camino hasta que los navegadores implementen este lenguaje de forma nativa, como lo vienen haciendo con JavaScript desde el principio. También debemos recordar que este no es el primer intento de cambiar el uso de JavaScript por otros lenguajes (no de la mano de Google). Los anteriores intentos no tuvieron éxito, incluso contra propuestas por parte de Microsoft. Por todo esto, deberemos esperar un tiempo antes de estar seguros si deberíamos empezar a mudar nuestras aplicaciones hacia este lenguaje. **P**

JUGANDO CON CSS3

Por Gabriel Gaitano Ornia

gabriel.gaitanoornia@pixelscode.com

La nueva versión de hojas de estilo proporciona a los diseñadores una renovada oferta de trucos y posibilidades a la hora de dar más brillo a sus páginas. Podemos ir viendo algunas de esas posibilidades del modo más agradable, con ejemplos que nos muestren cómo usarlas y nos llenen de ideas para empezar a “meter mano” en el código. Y, para empezar, podemos

analizar una de las novedades que brinda la nueva especificación, el uso de pseudo clases.

DE CLASES, ESTADOS Y PSEUDO CLASES

En muchos elementos dentro de HTML vamos a encontrarnos con estados especiales o usos vinculados. Es posible utilizando CSS aplicarles estilos diferentes a estos elementos

tomando como referencia estos estados. Vemos así en un ejemplo que el elemento `a` tiene cuatro estados:

- Normal: Estado original.
- Active: Este es el estado que corresponde al click del usuario.
- **Hover**: El cursor del mouse se haya sobre el elemento.
- **Visited**: El link al que hace referencia ya fue visitado.

Ahora bien, la pseudo clase es un uso predefinido o estado de un elemento HTML al que le podemos aplicar un estilo, sea cual sea su estado por defecto. Hay cuatro tipos diferentes de pseudo clases disponibles:

- **Estructurales:** Brindan la posibilidad de dar estilo a los elementos que deseemos a partir de una posición numérica exacta atribuida al elemento.
- **Links:** Las utilizamos para dar estilo al enlace, tanto en su estado normal por defecto como en cualquiera de sus estados (como vimos arriba).
- **Dinámicas:** Es posible aplicarlas a cualquier elemento para definir cómo se muestran cuando pasan entre distintos estados, como por ejemplo cuando estamos haciendo click en ellos, o si el cursor está situado por encima, o bien cuando son seleccionados por el usuario.
- **Otras generales:** Algunos elementos pueden recibir estilos de modo diverso, basándonos en el lenguaje o qué tipo de etiqueta no son.

Podemos verlo funcionando con una lista no ordenada:

```
<ul id="prueba">
  <li>Elemento inicial</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
  <li>Elemento 4</li>
</ul>
```

Ya tenemos el código HTML, ahora vamos a escribir el CSS:

```
li:first-child { font-size:
.9em; }
li:first-of-type { color:
blue; }
li:nth-of-type(4) { font-
```

```
size: 2em; }
li:nth-last-of-type(3) {
font-size: 1.5em; }
li:last-child { color:
red; }
```

Esto debe de renderizar los elementos de la lista de menor a mayor tamaño de fuente; aparte el primer elemento será azul y el último rojo, como se muestra a continuación:

Elemento inicial
Elemento 2
Elemento 3
Elemento 4

BORDEANDO LAS CURVAS

Con CSS3 podemos crear bordes redondeados con sólo aplicar propiedades de estilo sin utilizar un programa de diseño, como por ejemplo Photoshop. Vamos a ver la propiedad que lo permite: border-radius.

Esta caja tiene bordes redondeados.

Su aceptación no es universal, pero con algunas adiciones todos los navegadores lo permiten. Por ejemplo, Firefox exige el prefijo -moz- delante del nombre de la propiedad, como vemos en el siguiente ejemplo:

```
<html>
<head>
```

```
<style type="text/css">
div
{
border:2px solid #a1a1a1;
padding:10px 40px;
background:#dddddd;
width:300px;
border-radius:25px;
```



La pseudo clase es un uso predefinido o estado de un elemento HTML al que le podemos aplicar un estilo, sea cual sea su estado por defecto."

```
-moz-border-radius:25px;
/* Firefox 3.6 y posterior
*/
}
```

```
</style>
</head>
<body>
```

```
<div>Esta caja tiene
bordes redondeados.</div>
```

```
</body>
</html>
```

Y este es el resultado:

“Con CSS3 podemos crear bordes redondeados con sólo aplicar propiedades de estilo sin utilizar un programa de diseño, como por ejemplo Photoshop.”

“Una divertida posibilidad que ofrece CSS3 es agregar a nuestras páginas sombras a las cajas de texto, posibilidad que las aplicaciones de escritorio brindaban desde hace mucho.”

COLOREANDO LA WEB

Los colores son mostrados en las páginas a través de una combinación codificada, combinando los colores base Rojo, Verde y Azul.

¿Y cómo se manifiesta esa codificación? Los colores CSS son definidos usando una notación hexadecimal para las combinaciones de los colores base que mencionamos anteriormente (RGB). El valor más bajo que puede ser dado a estos colores es 0 (en hexadecimal 00), mientras que el valor máximo es 255 (hexadecimal FF).

A su vez, estos valores hexadecimales son mostrados como 3 números de dos dígitos comenzando con un signo #.

Aquí usamos un valor hexadecimal.

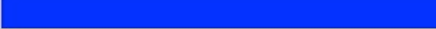
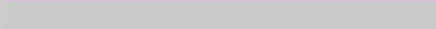
```
</p>
<p style="background-color:rgb(255,255,0)">Este color lo seteamos con un valor rgb.
</p>
```

```
<p style="background-color:yellow">
  Un método más clásico, con el nombre del color.
</p>
```

```
</body>
</html>
```

Y el resultado que podemos obtener así es:

Aquí usamos un valor hexadecimal. Este color lo seteamos con un valor rgb.

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

Veamos un ejemplo:

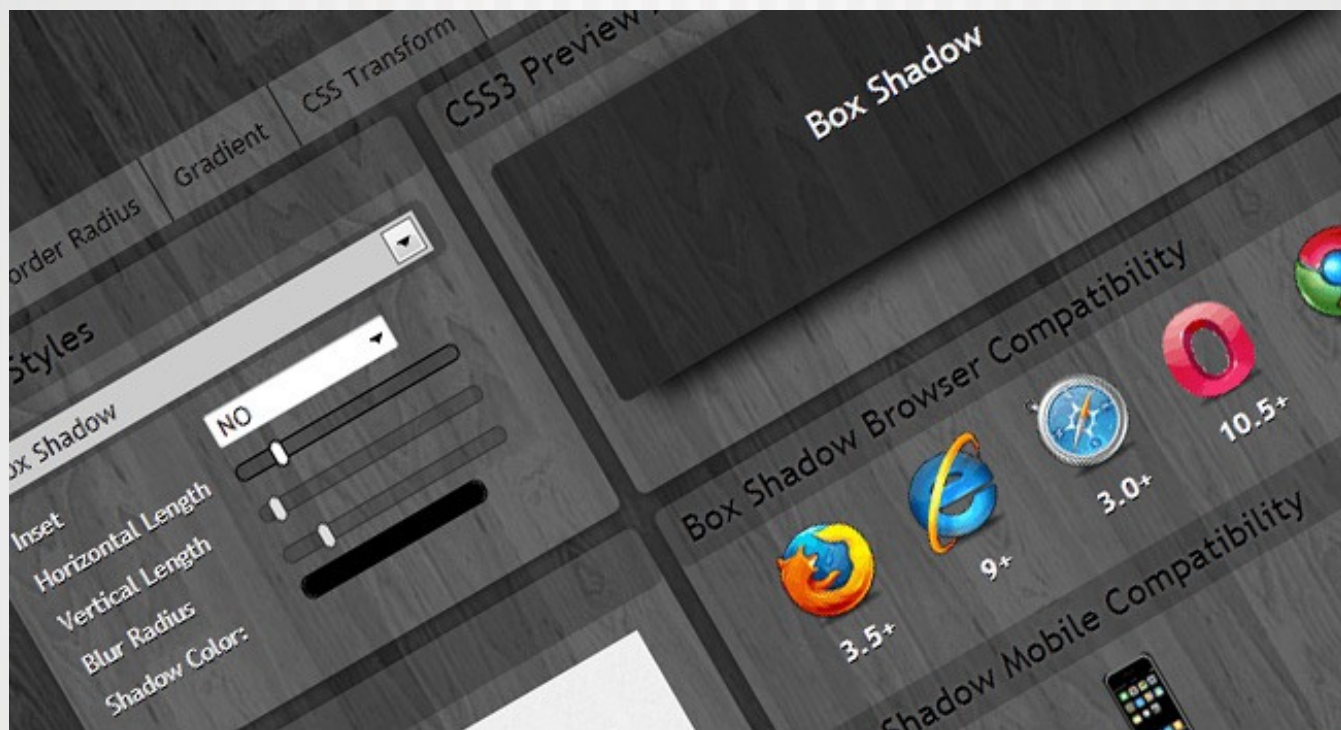
```
<html>
<body>

<p style="background-color:#FFFF00">
```

Un método más clásico, con el nombre del color.

ANIMÁNDONOS

Con CSS3 podemos crear animaciones sin recurrir a Flash, o imágenes



animadas o javascript en la mayor parte de los casos. Para esto es clave el uso de la regla `@keyframes`. Esta es la regla donde la animación es creada. Si especificamos un estilo CSS dentro de la regla `@keyframes`, la animación realizará un cambio gradual entre el estilo corriente y el nuevo por nosotros deseado.

Como suele pasar, el soporte para esta habilidad aún no es uniforme. Internet Explorer y Opera no soportan la regla `@keyframes` o la propiedad `animation`. Por el contrario, Firefox la admite precediéndola con el prefijo `-moz-`, mientras Chrome y Safari usan a su vez el prefijo `-webkit-`.

Veamos un ejemplo (en los navegadores que lo permitan):

```
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:100px;
background:red;
animation:myfirst 5s;
-moz-animation:myfirst 5s;
/* Firefox */
-webkit-animation:myfirst
5s; /* Safari and Chrome
*/
}
```

```
@keyframes myfirst
{
from {background:red;}
to {background:yellow;}
```

```
to {background:yellow;}
}
```

```
@-moz-keyframes myfirst /*
Firefox */
{
from {background:red;}
to {background:yellow;}
}
```

```
@-webkit-keyframes myfirst
/* Safari y Chrome */
{
from {background:red;}
to {background:yellow;}
}
```

```
</style>
</head>
<body>
```

>NOTA CENTRAL-



```
<p><b>Atención:</b> Este  
ejemplo no funcionará en  
Internet Explorer y Ope-  
ra.</p>
```

```
<div></div>
```

```
</body>  
</html>
```

El resultado que veremos será una animación de colores a nuestro DIV de rojo a amarillo, en suave cambio.

NO ME HAGAN SOMBRA

¿Cajas de texto sin sombra? ¡Qué antigüedad! Una divertida posibilidad que ofrece CSS3 es agregar a nuestras páginas sombras a las cajas de texto, posibilidad que las aplicaciones de escritorio brindaban desde hace mucho. Y, como dicen, para muestra basta un botón, aunque aquí veremos más bien texto:

```
<html>  
<head>  
<style type="text/css">  
h1  
{  
text-shadow: 5px 5px 5px  
#FF0000;  
}  
</style>  
</head>  
<body>
```

```
<h1>Efecto Sombra en nues-  
tro texto!</h1>
```

```
<p><b>Nota:</b> Internet  
Explorer no soporta la  
propiedad text-shadow.</p>
```

```
</body>  
</html>
```

Podemos especificar la sombra horizontal y la vertical, la distancia de difuminación de la sombra y su color.

Y el resultado se vería así:

Efecto Sombra en nuestro texto!

Nota: Internet Explorer no soporta la propiedad text-shadow.

ALINEANDO TEXTO

En CSS clásico no teníamos manera alguna de, si una palabra excedía a su contenedor, mantenerla o alinearla en los límites del área deseada.

Afortunadamente, CSS3 nos proporciona una nueva propiedad llamada word-wrap, la cual es soportada por todos los navegadores más conocidos y permite alinear debidamente el texto. Veamos cómo sería esto:

```
<html>  
<head>  
<style type="text/css">  
p.test  
{  
width:11em;  
border:1px solid #000000;  
word-wrap:break-word;  
}  
</style>  
</head>  
<body>
```

```
<p class="test"> Este pa-  
rrafo contiene una palabra  
muy larga: estaesunapala-  
brarealmente muy larga. La  
palabra larga sera cortada  
y se reiniciara en la li-  
nea siguiente.</p>
```

```
</body>  
</html>
```

Y lo que veríamos sería así:

Este párrafo contiene una palabra muy larga: estaesunapalabrarealmente muy larga. La palabra larga sera cortada y se reiniciara

Así nos permitiría ajustar con certeza el tamaño de nuestro texto al área deseada.

ROTANDO

A través de una propiedad nueva que nos ofrece CSS3 podemos aplicar transformaciones a un elemento. Una de las más interesantes aplicaciones de esta propiedad, muy correctamente llamada transform, es permitir rotar un elemento, aunque no es la única posibilidad. La propiedad transform es aceptada por todos los mayores navegadores, pero sólo con un prefijo delante que es -ms-transform en Internet Explorer, -moz-transform en Firefox, -o-transform en Opera y -webkit-transform para Safari y Chrome. Veámoslo funcionar:

```
<html>
<head>
<style type="text/css">
div
{
width:200px;
height:100px;
background-color:yellow;
/* Rotate div */
transform:rotate(7deg);
-ms-
transform:rotate(7deg); /*
Internet Explorer */
-moz-
transform:rotate(7deg); /*
Firefox */
```



CSS3 nos proporciona una nueva propiedad llamada word-wrap, la cual es soportada por todos los navegadores más conocidos y permite alinear debidamente el texto."

```
-webkit-
transform:rotate(7deg); /*
Safari and Chrome */
-o-transform:rotate(7deg);
/* Opera */
}
</style>
</head>
<body>
<div>Rotando</div>
</body>
</html>
```

Lo cual nos permitirá observar lo siguiente:



LOS PRO Y LOS CONTRA DE CSS3

En la etapa de desarrollo en la que nos encontramos, podemos ver muchas ventajas en el desarrollo utilizando los más nuevos estándar

res, como ser CSS3. Una interesante y variada oferta de nuevas propiedades y posibilidades, no siendo menor la de contar con una única herramienta estandarizada para el diseño. En un futuro esto afectará las posibilidades de otras herramientas, como ser Flash, pero aun así, y a despecho de los seguidores de este producto, no nos cabe duda de que el cambio será positivo.

¿Cuál puede ser la serpiente en nuestro paraíso? Sin sombra de dudas el hecho que la gran masa de usuarios es lenta en adoptar navegadores modernos, con la más amplia cobertura de los estándares ofrecidos. Y, en grado superlativo, las empresas u organizaciones que están detrás de esos navegadores y que no suelen trabajar de un modo muy unificado. Ya lo hemos visto en los ejemplos, tenemos desde propiedades con nombres adaptados, hasta otras directamente no soportadas.

Sin embargo, es posible imaginar que en un tiempo que nos parecerá sin duda demasiado largo irá ampliándose al mismo tiempo la base de aceptación y cumplimiento de los estándares, y esto nos permitirá disfrutar de las capacidades de CSS3 en todo su esplendor, como hemos empezado a mostrarlo en los ejemplos que vimos.

Para terminar, veamos algunos links donde podemos obtener más información, que nos han servido de excelente fuente y referencia:

<http://www.w3.org/TR/CSS/#css3>
<http://www.w3schools.com/default.asp>
<http://skyje.com/2010/11/css3/>
<http://www.catswhocode.com/blog/8-examples-of-stunning-css3-text-effects> **P**



Por Matías Iacono,
matias.iacono@pixelscode.com

En el artículo anterior sobre plantillas T4 aprendimos a crear código fuente de forma dinámica; el mismo es incorporado automáticamente a nuestro proyecto y queda listo para ser usado. En este artículo aprenderemos otras técnicas que pueden aplicarse al uso de estas plantillas, como el uso de código externo, o la ejecución y captura del resultado de la plantilla sin la necesidad de generar un archivo de código para nuestro proyecto.

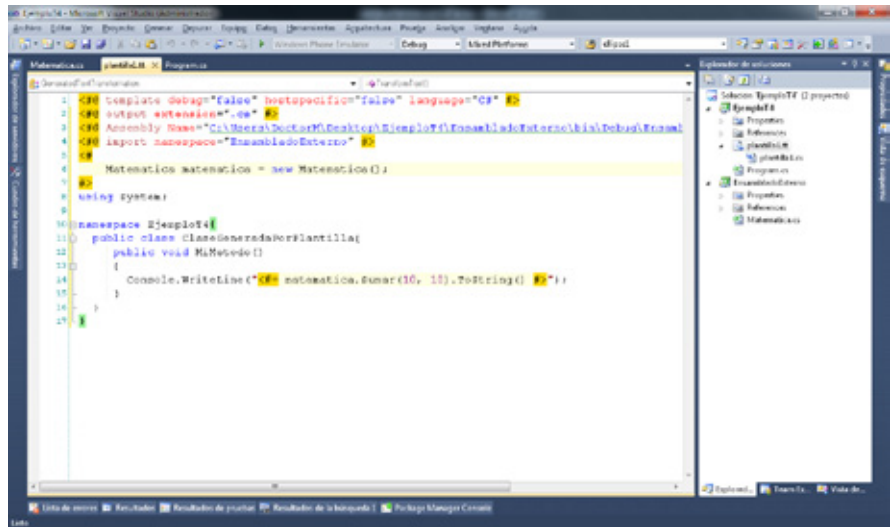
USANDO LIBRERÍAS EXTERNAS

Además de ejecutar el código de la plantilla con valores y funciones creadas dentro de estas es posible utilizar código externo que nos provea tanto de funcionalidad como de datos. De esta forma es posible tener reglas de negocio en ensamblados externos a nuestro proyecto y utilizar su funcionalidad para construir una plantilla y su resultado.

```
namespace EnsambladoExter-
```

```
{
    public class Matema-
    tica
    {
        public int
        Sumar(int valor1, int va-
        lor2)
        {
            return valor1
            + valor2;
        }
    }
}
```

El código anterior representa un ensamblado externo al proyecto que



```
namespace="EnsambladoExterno"
#>
<#
    Matematica matemati-
ca = new Matematica();
#>
using System;
```

```
namespace EjemploT4{
    public class ClaseGene-
radaPorPlantilla{
        public void MiMeto-
do()
        {
            Console.WriteLi-
ne("<#= matematica.Su-
mar(10, 10).ToString()
#>");
        }
    }
}
```

Mediante la directiva `<#@ Assembly ... #>` es posible especificar la ruta y el nombre del ensamblado que pretendemos usar dentro de la plantilla. Luego, mediante código estándar será posible utilizar los objetos contenidos en éste y aplicarlos a la generación de código. En nuestro ejemplo, sumamos dos

El ensamblado externo debe ser compilado antes de poder ser usado en la plantilla T4.

números y asignamos su resultado a la línea que escribiría un valor en la consola.

```
using System;
namespace EjemploT4{
    public class ClaseGene-
radaPorPlantilla{
        public void MiMeto-
do()
        {
            Console.WriteLi-
ne("20");
        }
    }
}
```

Si inspeccionamos el resultado de la plantilla podremos ver que la ejecución de la función `Sumar` se ha ejecutado y el resultado escrito en el código.

TEXTO PRE PROCESADO
Otra cualidad útil de las plantillas

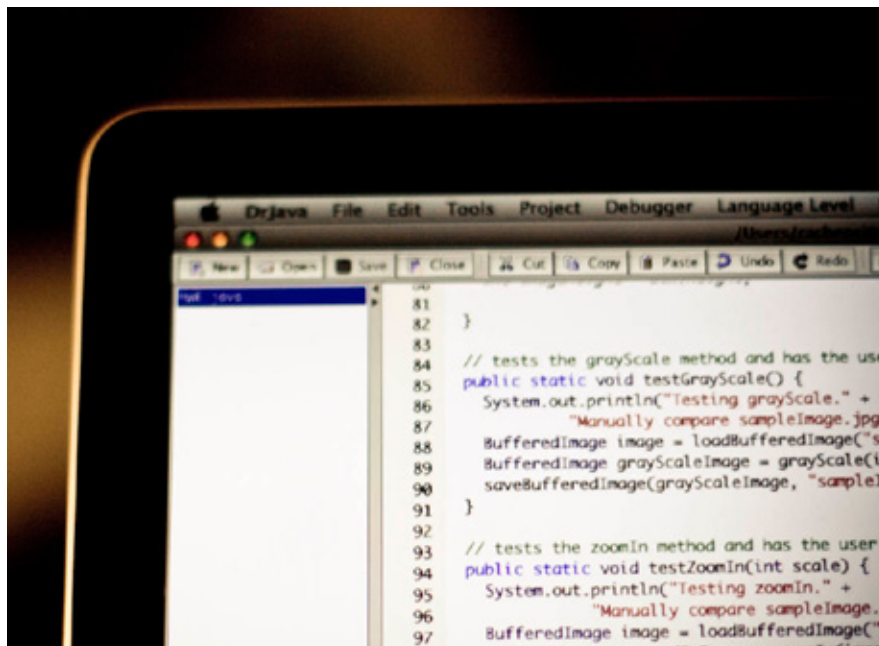
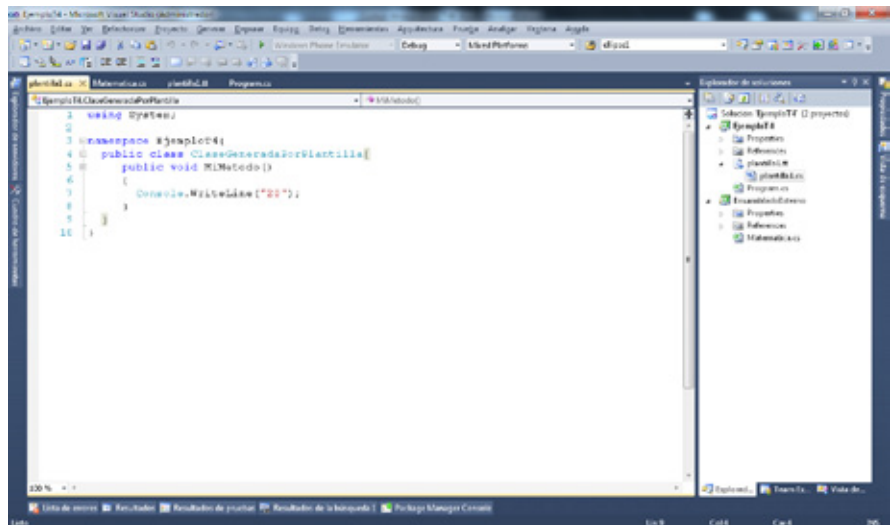
contiene la plantilla T4. En este se encuentran ciertas reglas de negocios como las operaciones matemáticas. Si bien el ejemplo puede resultar simple, es útil para ilustrar el comportamiento de la plantilla. Por lo tanto, teniendo este componente compilado, desde la plantilla podremos tomar el mismo e incorporarlo a la lógica de ejecución y generación de código.

```
<#@ Assembly Name="[RUTA
DEL ENSAMBLADO]\Ensambla-
doExterno.dll" #>
<#@ import
```

>PROGRAMACIÓN-



Es posible tener reglas de negocio en ensamblados externos a nuestro proyecto y utilizar su funcionalidad para construir una plantilla y su resultado."

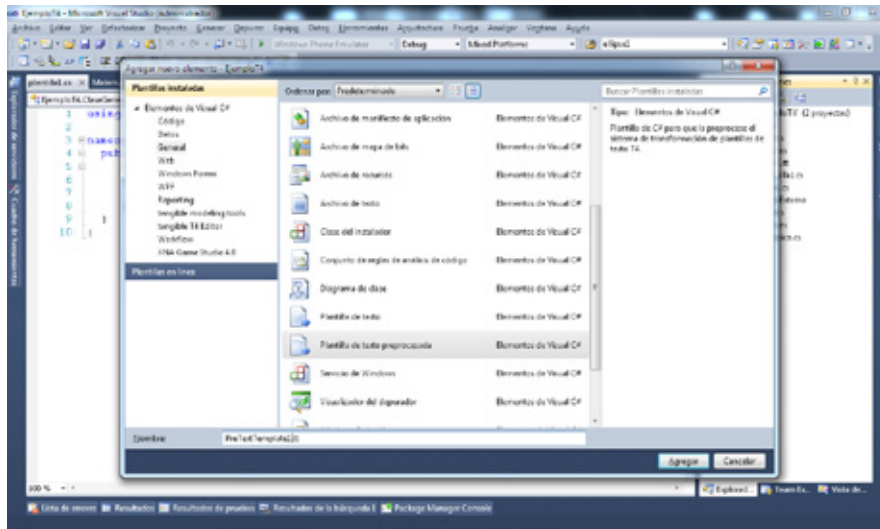


Todo el código es creado. En el momento de la creación las funciones contenidas en la plantilla se ejecutarán y sus resultados serán agregados al código final.

Esto quiere decir que podríamos generar archivos de texto, páginas HTML, archivos XML o código C# aunque sin importar el formato o tipo de elemento que estemos creando, siempre será tratado como texto.

Para crear una plantilla para texto pre procesado, además de crear la plantilla, deberemos modificar la herramienta de procesamiento que la plantilla usará para ejecutarse. Desde las opciones de la plantilla podremos modificar la propiedad Herramienta Personalizada para utilizar la TextTemplatingFilePreprocessor (preprocesador de archivo de plantilla de texto), al colocar este

T4 es la posibilidad de no generar código o archivos en tiempo de compilación o guardado si no en el momento en que nosotros decidamos. La diferencia principal es que el resultado que obtendremos será texto plano sin importar el tipo de texto que queramos crear.



Las plantillas pre procesadas tienen la misma nomenclatura pero se diferencian por el tipo de interprete que utilizarán.

```
{
    var t = new Pre-
    TextTemplate1();
    Console.
    WriteLine(t.TransformText());
}
```

atributo la plantilla ya no se compilará en el momento que la estemos diseñando sino que esperará a que le digamos, desde nuestro código, de forma explícita, que debe hacerlo.

```
<#@ template language="C#"
#>
<#@ assembly name="System.
dll" #>
namespace EjemploT4
{
    public class ClasePre-
    ProcesadaDeTexto{
        public void Escri-
        bir()
        {
            Console.
            WriteLine("Hola Mundo");
        }
    }
}
```

Como podemos ver, el código de la plantilla anterior no difiere de las creadas hasta el momento. La diferencia estará en la forma en cómo

```
_files = array();
ed_files = array();
ate_log as $update_log_rev)

update_log_rev['paths'] as $update_log_path

that are part of the same revisions in the
($update_log_path['path'], 0, strlen($upd

date_log_path['action'] == 'M' || $update_
date_log_path['action'] == 'A') { $update_
date_log_path['action'] == 'D') { $update_
```

la ejecutemos. De esta forma, desde nuestro código podremos hacer lo siguiente para llamar a esta plantilla y ejecutarla.

```
class Program
{
    static void
    Main(string[] args)
```

```
Console.ReadLi-
ne();
}
}
```

En el código anterior, primero creamos un nuevo objeto del tipo de nuestra plantilla. Mediante la función TransformText (TransformText)

>PROGRAMACIÓN-

mar texto), la plantilla se ejecutará, creará el resultado contenido en ella y nos lo devolverá para que lo manipulemos en base a nuestras necesidades.

PASO DE PARÁMETROS

Otra característica importante es la posibilidad de pasar parámetros a una plantilla desde nuestro código. Esto nos posibilitará, al igual que la ejecución de funcionalidad contenida en ensamblados externos, moldear el resultado de la plantilla en base a posibles interacciones con el usuario.

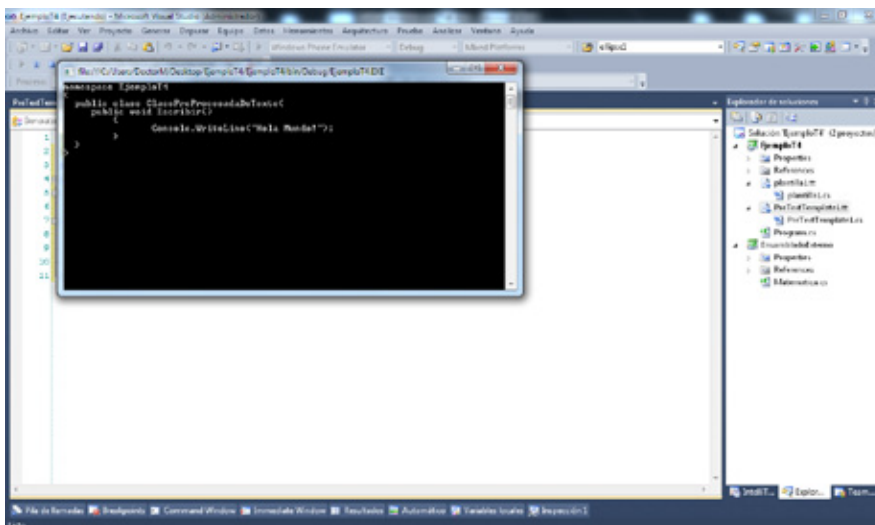
```
...  
public partial class Pre-
```

Al ejecutar el código el valor de la cadena de texto es incluido en el resultado generado por la plantilla.

```
TextTemplate1  
...
```

Si inspeccionamos la clase asociada a la plantilla veremos que la misma está declarada como Partial (Parcial), esto quiere decir que la clase podría estar contenida en más de un archivo físico y por lo tanto nos permitirá crear más funciona-

Al ejecutar la aplicación, la plantilla nos retorna el resultado de la misma procesada.



“Las plantillas T4 poseen la flexibilidad suficiente para poder ayudarnos a construir generadores de código fuente, así como elementos de texto claves para el funcionamiento de nuestras aplicaciones.”

```

compares the first result to see if it contains
e = True
for n = 0 To 7
    anagram = True
    For m = 0 To numlen - 1
        'Console.WriteLine("Comparing " & arrNumbers(n).
        If InStr(arrNumbers(n).temp, arrNum(m)) <> 0 Then
            InNum = True
            arrNumbers(n).temp = Left(arrNumbers(n).temp,
            ' Console.WriteLine(arrNumbers(n).temp)
        Else
            InNum = False
            anagram = False
        End If
        'Console.WriteLine(InNum)
    Next
    'Console.WriteLine("Anagram = " & anagram)
    If anagram = True Then
        Console.WriteLine(arrNumbers(n).temp)
    No = False
    End If
Next

```

lidad que se adicionará a la clase ya establecida. Una de las formas de poder pasarle parámetros a una plantilla es por medio de propiedades en la clase que representa la misma.

```

public partial class PreTextTemplatel
{
    public string ValorAMostrar { get; set; }
}

```

En nuestro código adicionamos una propiedad del tipo string que contendrá el valor que queramos pasarle a la plantilla desde el código en ejecución. Como la plantilla es declarada como un

objeto desde el código, mientras este esté en funcionamiento, el valor contenido en la propiedad será accesible desde el código de la plantilla.

```

<#@ template language="C#"
#>
<#@ assembly name="System.
dll" #>
namespace EjemploT4
{
    public class ClasePreProcesadaDeTexto{
        public void Escribir()
        {
            Console.WriteLine("<# = ValorAMostrar
#>");
        }
    }
}

```

```

}
}

```

En la plantilla, entonces, con sólo acceder al valor de la propiedad incluiremos éste en el resultado de la misma una vez sea procesado. Por supuesto, necesitaremos configurar este valor antes de obtener el resultado de la plantilla.

```

static void Main(string[]
args)

```

```

{
    var t = new PreText-
    Templatel();
    t.ValorAMostrar =
    "Hola Mundo!";
    Console.WriteLine(t.
    TransformText());
    Console.ReadLine();
}

```

RESUMEN

Las plantillas T4 poseen la flexibilidad suficiente para poder ayudarnos a construir generadores de código fuente, así como elementos de texto claves para el funcionamiento de nuestras aplicaciones. Posiblemente, un punto negativo sea el código espagueti que obtendremos en la misma plantilla, combinando código ejecutable con texto que maqueta el código final. De cualquier manera, acostumbrarse a su uso no será difícil y los resultados pueden aumentar fácilmente nuestra productividad y eficiencia en la creación de código. **P**

Coworking: Tu free-office

Por Roxana Miguel

roxana.miguel@pixelscode.com

@roxanamiguel



Los freelancers ahora cuentan con un espacio para sacar el trabajo de sus casas y disfrutar de todas las comodidades de una oficina sin perder la independencia.

Si estás pensando en cambiar tu teléfono por un conmutador, si tu heladera tiene tantas o más anotaciones que una agenda, si tu notebook está siempre encendida y si tu mesa de luz tiene un listado de tareas, lapiceras, post-it y elementos de escritorio encima, es señal de que tu casa se ha convertido en tu empresa y no hay rincón en

ella en donde puedas escapar de las obligaciones.

Este es el principal inconveniente de ser freelancers: prácticamente no tienen una frontera definida entre el tiempo privado y el laboral y, por ende, no sólo reciben el correo bajo la puerta del hogar, sino que las llamadas a cualquier hora y los



el país ubicados, en su mayoría, en Capital Federal, aunque también los hay en provincias como Formosa y Mendoza, y ciudades como Rosario y Mar del Plata.

No sólo se trata de un espacio para trabajar, sino también para recibir clientes, organizar reuniones e interactuar con pares. Son edificios amplios con lugares comunes donde compartir experiencias con colegas o conocer a otros profesionales. Las oficinas se alquilan y uno puede instalarse, pues allí encontrará todo lo necesario como para trabajar sin inconvenientes.

compromisos de reuniones siempre se hacen en el seno familiar.

De un tiempo a esta parte, los freelancers se han multiplicado y es porque para muchas compañías es conveniente tener una planta de tele trabajadores. Pensando en estas desventajas de los que trabajan desde sus casas es que desde hace

unos años se impuso una modalidad llamada coworking.

En la argentina comenzaron a construirse a partir del 2007 edificios cuya utilidad es brindar soluciones, justamente, a quienes quieren lograr esta división entre hogar-oficina. La metodología de Coworking ya generó al menos once centros en todo

Aquí no existe un jefe que indica la hora de entrada o salida, no hay que marcar tarjeta ni hay que pedir médico, ni poner excusas para no ir a trabajar; los horarios los elige uno mismo, al igual que los compañeros de trabajo, dado que en el escritorio cercano siempre tendremos a otros freelancers en búsqueda de lo mismo: un espacio para trabajar.

“Aquí no existe un jefe que indica la hora de entrada o salida, no hay que marcar tarjeta ni hay que pedir médico, ni poner excusas para no ir a trabajar; los horarios los elige uno mismo, al igual que los compañeros de trabajo, dado que en el escritorio cercano siempre tendremos a otros freelancers en búsqueda de lo mismo: un espacio para trabajar.”

AMBIENTES **DESCONTRACTURADOS**

Parecen cafés con wi-fi libre, porque hacia donde dirijamos las miradas hay personas con sus notebooks sentadas a la mesa. La diferencia es que están todos concentrados en sus labores y no hay mozos, ni barras de bebidas. Algunos están reunidos en unas salas apartadas, otros circulan entre las mesas para hacer una consulta.

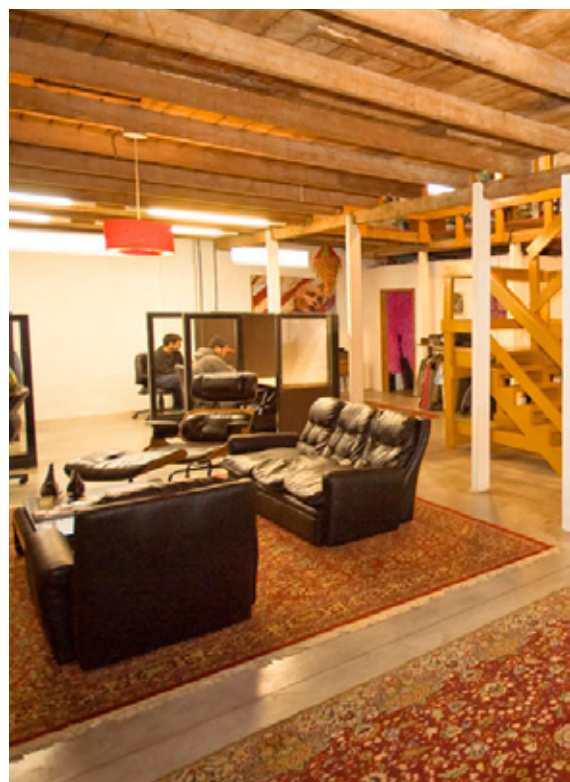
Las primeras oficinas coworking aparecieron en Capital Federal en los barrios Palermo y Belgrano; al poco tiempo empezaron a abrir sus puertas otros proyectos similares y todos con un mismo objetivo: ofrecer un ambiente descontracturante en donde trabajar.

“Nuestro primer local lo abrimos en 2009. Tomando como punto de partida la evolución de la forma de trabajar y detectando una necesidad no cubierta, creamos este espacio diseñado para los trabajadores móviles que combina la informalidad de un bar con todos los servicios de una oficina,” nos cuenta Florencia Faivich, socia de Urban Station (argentina. enjoyurbanstation.com). “Actualmente ofrecemos un sistema por el cual les prestamos bicicletas a los clientes para que puedan cortar con su rutina”. El éxito de esta experiencia les permitió abrir cuatro locales, dos en Buenos Aires, uno en Mendoza, otro en Chile y, próximamente, en Bogotá.

Estos espacios nacieron en 2005 originariamente en San Francisco, Estados Unidos, por voluntad de un grupo de personas que se reunían en algunos cafés o en sus propias casas para trabajar. Con el tiempo, esta modalidad se fue consolidando hasta lograr una red en todo el mundo de lugares donde se pueden crear sinergias entre personas que participan regularmente.

Uno de los lugares recientemente abiertos en Capital Federal es Espacio Céspedes (www.espaciocespedes.com), que empezó a funcionar en julio de este año. “La idea surgió a partir de una necesidad propia en un emprendimiento anterior, viendo la importancia del contacto con grupos más amplios, diversos y la energía creativa que esto genera. Esta necesidad se combinó con mi interés por el mercado inmobiliario y la arquitectura





para encontrar una buena oportunidad y diseñar un espacio pensando en los detalles para lograr una propuesta de calidad," nos cuenta Paul Iribarne, fundador de Céspedes.

En Céspedes se reúnen personas de distintas disciplinas, "se fue consolidando el primer grupo de usuarios con perfiles que trabajan en diseño gráfico, arquitectura, diseño web, periodismo, comunicación sustentable, marketing online y distribución independiente de cine". Por su parte, Florencia nos cuenta que en el caso de Urban Station "el público es bastante heterogéneo: desde contadores, consultores en RRHH, diseñadores y fotógrafos. Pero muchos de nuestros clientes tienen que ver con el mundo de la tecnología (programadores, diseñadores web, etc.), quienes encuentran en nuestro

espacio la opción ideal para llevar adelante sus operaciones y acelerar su crecimiento".

La conjunción de distintos profesionales en áreas comunes permite que muchos se consulten temas entre sí, o se busquen a través de las redes para colaborar con proyectos personales. Es así como los coworking se convierten en pulmones laborales donde cada integrante, aunque esté alquilando por temporada o lo haga esporádicamente, puede transformarse en una pieza fundamental en el trabajo de otro. Esa sinergia es la magia que hace que estos lugares funcionen tan bien.

¿CÓMO SER COWORKERS?

Necesitando un espacio. Cuando las particularidades nombradas al comienzo del artículo se hacen

notar en la vida de un freelancer, entonces es hora de tomar el toro por las astas y poner a prueba esta experiencia. En la mayoría de las oficinas coworking la organización es casi similar: se puede contratar el servicio por día, por semana o por mes.

En el caso de Céspedes ofrecen distintos planes: Full, Flexible, Pack Horas o Pase diario. Cada uno brinda diferentes tipos de servicios mobiliarios, cajonera con llave, sala de reuniones, locker, teléfonos o coffe store. Mientras que Urban Station ofrece Labtops Locks, diarios y revistas, oficina virtual, servicio de motos y hasta cargadores para celulares. Los valores varían según lo que requiera el coworker, el lugar donde va a trabajar y la privacidad que necesita para ello. En este sentido, Paul Iribarne nos



agrega que “la respuesta ha sido muy buena en términos cuantitativos y cualitativos, con mucho aprendizaje en los meses iniciales. Hasta el momento, se han sumado 14 personas a nuestra propuesta, con esquemas que van desde un puesto fijo en el plan “full” hasta packs de horas. También hemos sumado usuarios que usan exclusivamente la sala de reuniones.”

La filosofía del coworking pareciera estar comandada por el término “compartir” ya que además de oficinas, livings, salas de reuniones, pasillos y patios comunes vienen bien para enriquecer la experiencia, también se disponen cocinas totalmente equipadas con utensillos para tomar un break y distenderse unos minutos antes de continuar.

“Esto es un fenómeno que está pasando, una evolución de la forma de trabajar que tiene un impacto económico, social y cultural”, comentó hace

unos años Claudio Bisurgi, entonces socio de Urban Station, al diario La Nación cuando fue consultado por la novedad de los coworking. En aquel momento existían sólo tres oficinas en todo Buenos Aires, y era la apertura de una modalidad que desde el 2007 hasta entonces creció en todo el país.

LOS COWORKING ABREN SUS PUERTAS

A raíz de la gran demanda de espacios para coworkers, muchos se han visto motivados a incursionar en estos emprendimientos. Conozcamos otros espacios:

Workstation: El edificio está ubicado en el barrio de Belgrano de Capital Federal. Comenzó a funcionar en 2010 orientado a trabajadores independientes y empresas. La segunda opción está dirigida a las compañías que no cuentan con capacidad edilicia para reuniones.

Brindan oficinas privadas, secretaria, recepción de correo postal y envío de correspondencia, Internet (Wi-Fi y plug in), sala de reuniones con multimedia, servicio de impresiones y escaneos.

www.workstationba.com

Coworking Buenos Aires: Ubicado en Palermo Soho, el espacio creado por Favio Trasi, resultó de un negocio anterior. Ofrece un lugar para freelancers y grupos de trabajadores de forma permanente si lo requieren. Las oficinas cuentan con la ambientación y el amoblamiento necesarios, teléfonos internos, Internet, sala de reuniones a disposición, cocina y recepción de clientes.

coworkingbuenosaires.com

Loft Coworking: Esta es la primera oficina compartida de Mar del Plata, está dirigida por Francisco Mas, quien se adjudica la creación del

lugar debido a que su trabajo estaba dirigiéndose en el mismo sentido que la tendencia que se estaba dando por entonces en Capital Federal. Loft ofrece espacios individuales y áreas de trabajo común, cocina equipada y un living con cómodos sillones.

www.loftcoworking.com.ar

Njambre: En la ciudad santafesina de Rosario abrió sus puertas en abril de 2011 este espacio comandado por Federico Seineldín, pensado para que la comunidad de emprendedores de negocios, creativos, freelancers, ONGs y pequeñas empresas compartan el día a día. A diferencia de otros, Njambre hace mucho hincapié en la sustentabilidad del proyecto como una forma de favorecer al medio ambiente desde la tarea cotidiana de trabajar, y al mismo tiempo compartir conocimientos y capacidades. Ofrece espacios abiertos, tarjeta magnética de ingreso, lockers, sala de reuniones, telefonía, cocina, playroom, biblioteca, patio deck y café, mate, agua y frutas sin cargo.

www.njambre.org

Distintos países de Latinoamérica y España se han sumado a la propuesta abriendo oficinas compartidas para todos los gustos. Es así como Urban Station, además de echar raíces en Argentina, también lo está haciendo en Chile y Colombia. En Panamá abrió sus puertas The Casco Station, otro emprendimiento similar y ColectivoLabs en México.

Los freelancers viajeros cuentan con directorios para rastrear sus propios coworkings en donde estén. En coworkingspain.es/ encontrarán sitios de España, coworkingbrasil.ning.com/page/espacos-de-coworking, en Brasil o bien loosecubes.com/ en cualquier parte del mundo.

Combatir el aislamiento, encontrar un espacio de interacción con otros que posibilite potenciar nuestro trabajo y nuestras ideas, compartir con profesionales nuestros proyectos para recibir sus apreciaciones o correcciones, aprender a trabajar en equipo, llegar a casa y vivir la tranquilidad de un hogar separa-

“Aquí no existe un jefe que indica la hora de entrada o salida, no hay que marcar tarjeta ni hay que pedir médico, ni poner excusas para no ir a trabajar; los horarios los elige uno mismo, al igual que los compañeros de trabajo, dado que en el escritorio cercano siempre tendremos a otros freelancers en búsqueda de lo mismo: un espacio para trabajar.”

do de las obligaciones, esos son los beneficios de los coworkers. Afortunadamente, el coworking une lo mejor de trabajar en una oficina con lo mejor de trabajar en casa. **P**





Algoritmos: comb sort

Por Juan Gutmann
juan.gutmann@pixelscode.com

Cuando en el primer número de Pixels&Code analizamos desde esta sección el método de ordenamiento por burbujeo, o Bubble Sort, explicamos que se trataba de uno de los algoritmos más sencillos que existen, aunque la eficiencia de su implementación estándar (sin incorporar ningún tipo de optimizaciones) es bastante pobre. Seguramente nuestros lectores no pensaron en aquel momento que Bubble Sort, siendo tan básico, daría sin embargo tanta tela para cortar. Recientemente aprendimos el Gnome Sort, que se basa tanto en éste como en Insertion Sort. En esta ocasión vamos a ver otra variante del sistema de burbuja, conocido como Comb Sort, que demuestra cómo con un pequeño cambio se puede mejorar su performance considerablemente.

CONEJOS Y TORTUGAS

El lector atento seguramente recordará que al detallar los aspectos teóricos de Bubble Sort, mencionamos los “conejos” y las “tortugas”. Con el primer término nos referimos a los valores más altos de la lista, que se ubican en las primeras posiciones de la misma. Por la naturaleza misma del funcionamiento del burbujeo, los conejos son desplazados en las primeras pasadas hacia su posición correcta en el final de la misma. En cambio, los valores bajos que se encuen-

tren en las últimas posiciones (las tortugas), tardan una considerable cantidad de ciclos de procesamiento para llegar a su lugar correspondiente, al inicio de la lista. Se explica entonces el por qué de estos curiosos apelativos. A principios de la década del ochenta, el científico informático Włodzimierz Dobosiewicz, un especialista en sistemas de ordenamiento, se dedicó a estudiar a fondo el pequeño y sencillo algoritmo de burbujeo, y notó rápidamente que su talón de Aquiles era justamente la dificultad del mismo para lidiar con las “tortugas”. Este déficit lo hacía muy dependiente de las características de la lista a ordenar para lograr tiempos de respuesta razonables. Para lograr que su efectividad fuera más independiente del tipo de lista, como sucede con algoritmos estilo “divide and conquer” (por ejemplo, Quicksort), se propuso modificar Bubble Sort para que las lentas tortugas se desplacen más rápidamente.

EL “GAP”

La causa de la lentitud de las tortugas, que se ubican en el final de la lista, es que Bubble Sort compara cada elemento de la lista con el que le sigue, comenzando por el principio de la misma. Es decir, la distancia (o “gap”) entre los elementos que se comparan es siempre de una posición. Dobosiewicz observó que el algoritmo de ordenamiento propuesto por otro experto, David Shell, en 1959, y que lleva el nombre de su creador (“Shell Sort”), se basa en comparar elementos que se encuentran a una mayor distancia uno de otro. Por ende, concluyó, si comparamos el primer elemento de una lista con otro que se en-

cuentre casi en el final, y tras cada recorrida de la lista reducimos la distancia (gap) entre los elementos a comparar, podemos obtener una versión de Bubble Sort mucho más eficiente al procesar las tortugas, y por ende que se vea menos penalizada al tratar listas en las que los valores más bajos se encuentran en las últimas ubicaciones.

EL NÚMERO MÁGICO

El factor crítico de este enfoque es precisamente el valor del gap. Es fundamental para la eficiencia del algoritmo dar con los valores más adecuados tanto al comienzo del proceso como tras cada recorrida de la lista. Tras una serie de pruebas analíticas, Dobosiewicz propuso dar al gap inicial un valor equivalente a la mitad de la cantidad total de posiciones de la lista ($N/2$) y reducir el gap tras cada pasada, haciéndolo equivalente al valor entero más cercano al gap dividido por una constante numérica, que tras diversos experimentos estableció en 1,33. En un estudio posterior, llamado “A Fast Easy Sort”, escrito por Stephen Lacey y Richard Box y publicado en la legendaria revista “Byte” en 1991, los autores realizaron un exhaustivo trabajo estadístico para determinar los valores ideales tanto para el gap inicial como para el factor de reducción del mismo. Como resultado, fijaron el primero en $N/1,3$ y el “shrink factor” en 1,3. Posteriormente, otros investigadores establecieron que el factor de reducción más óptimo es la constante 1,247330950103979. Por razones prácticas, diversas implementaciones de Comb Sort utilizan redondeos de esta constante, como 1,25 o directamente 1,3.

VERSIÓN EN PSEUDOCÓDIGO

A continuación, el pseudocódigo correspondiente. Nótese que además de trabajar con el gap (que aquí llamaremos “M”), esta versión incorpora una característica similar a una de las dos optimizaciones más comunes para Bubble Sort: la detección de la ausencia de intercambios tras una pasada completa, lo que implica que la lista ya está totalmente ordenada, evitando subsiguientes iteraciones, ya innecesarias. En Comb Sort se verifica que la lista ya está ordenada cuando no tiene lugar ningún intercambio con un valor de distancia (gap) igual a 1.

Dada una lista L, de N elementos

```

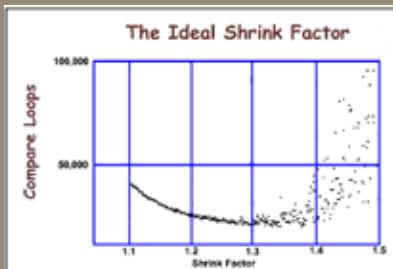
M = N
mientras no se dé que M = 1 e intercambio = falso
    M = M / 1.247330950103979
    redondear M al valor entero más próximo

    si M < 1
        M = 1 /* el gap (M) nunca puede valer menos que 1 */
    fin si

    I = 0
    intercambio = falso

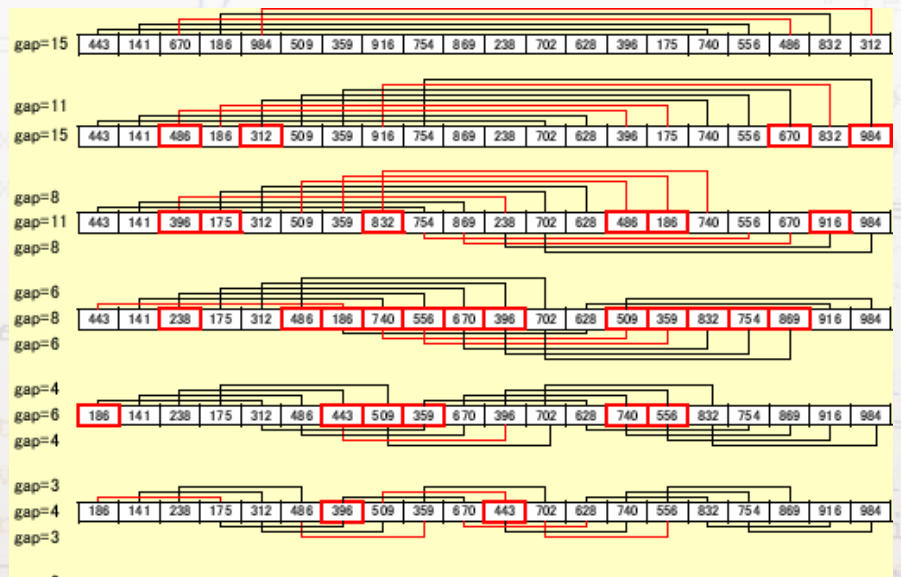
    mientras I + M < N
        si L[I] > L[I + M]
            intercambiar L[I] y L[I + M]
            intercambio = verdadero
        fin si
    fin si

```

Este gráfico ilustra el análisis estadístico realizado para determinar el factor de reducción del gap más eficiente.

“Si comparamos el primer elemento de una lista con otro que se encuentre casi en el final, y tras cada recorrida de la lista reducimos la distancia (gap) entre los elementos a comparar, podemos obtener una versión de Bubble Sort mucho más eficiente.”



Al ejecutar el código el valor de la cadena de texto es incluido en el resultado generado por la plantilla.

```
I = I + 1
fin mientras
fin mientras
```

IMPLEMENTACIÓN EN PYTHON

Como ustedes ya saben bien, a la versión en pseudocódigo le sigue una implementación en lenguaje Python, intentando mantener una estructura lo más similar posible al pseudocódigo. Esto nos permitirá ver el algoritmo en acción, algo particularmente útil en el caso de Comb Sort, ya que podemos analizar el impacto de las características de la lista sobre el tiempo de ejecución del mismo, y compararlo contra las distintas versiones de Bubble Sort para la misma lista.

```
# -*- coding: cp1252 -*-
```

```
def combsort(L):
```

```
itera = 0
N = len(L)
M = N
```

```
while not (M == 1 and
intercambio == 0):
    M = int(M /
1.247330950103979)
    if M < 1:
        M = 1
```

```
I = 0
intercambio = 0
```

```
while I + M < N:
    if L[I] > L[I +
M]:
        valor = L[I]
        L[I] = L[I +
M]
        L[I + M] = va
lor
        intercambio =
1
```



```

typedef struct tree_el node;

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

```

```

I = I + 1

itera = itera + 1

print "iteración =",
itera, "I =", I, "M =", M,
"-->", L

return L

Z = [5, 3, 9, 7, 10, 6, 4,
8, 2, 1]
print "Lista inicial", Z
print combsort(Z)

```

Para apreciar mejor la efectividad de Comb Sort a la hora de tratar las tortugas, hemos ubicado los valores más bajos al final de la lista. La salida por pantalla del script Python es la siguiente:

```

Lista inicial [5, 3, 9, 7, 10, 6, 4, 8, 2, 1]
iteración = 2 I = 2 M = 8
--> [2, 1, 9, 7, 10, 6, 4, 8, 5, 3]
iteración = 6 I = 4 M = 6
--> [2, 1, 5, 3, 10, 6, 4, 8, 9, 7]
iteración = 12 I = 6 M = 4
--> [2, 1, 4, 3, 9, 6, 5, 8, 10, 7]
iteración = 19 I = 7 M = 3
--> [2, 1, 4, 3, 8, 6, 5, 9, 10, 7]
iteración = 27 I = 8 M = 2
--> [2, 1, 4, 3, 5, 6, 8, 7, 10, 9]
iteración = 36 I = 9 M = 1
--> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
iteración = 45 I = 9 M = 1

```



```

--> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for(i=1;i<=10;i++)

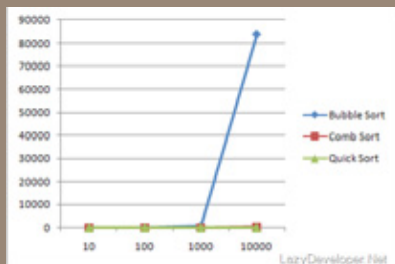
```

Como pueden apreciar, las “tortugas” se desplazan hacia las posiciones iniciales de la lista prácticamente en las primeras pasadas, en lugar de tener que esperar a las últimas iteraciones, como sucede en la implementación tradicional del método de burbuja. Por razones de espacio, en esta oportunidad optamos por imprimir el estado de la lista solamente al finalizar cada pasada del bucle externo. Para ver más en detalle los cambios tras cada iteración, pueden agregar un nivel de indentación a la línea que contiene el comando “print”, mo-

viéndolo de esta forma dentro del bucle interno.

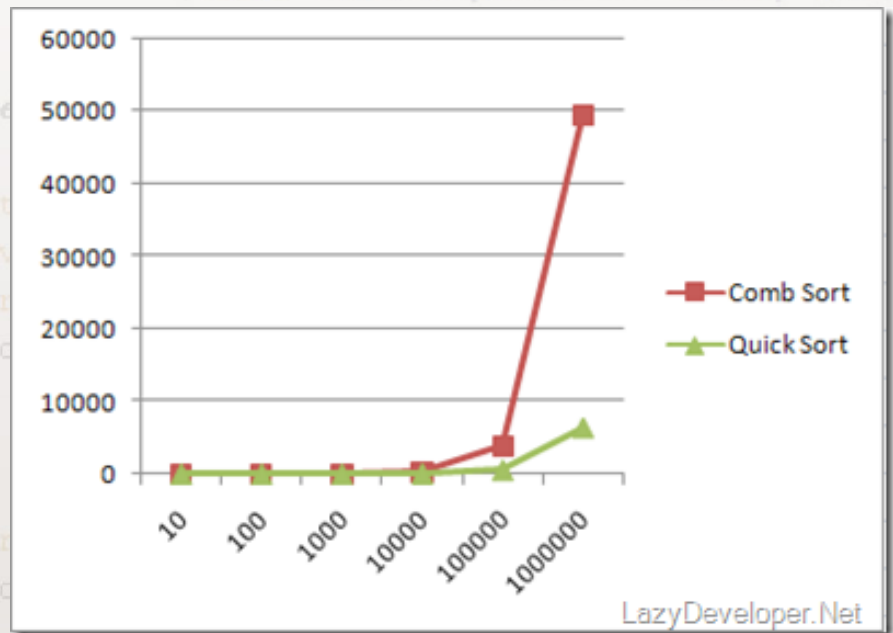
OPTIMIZACIONES

La optimización más popular para Comb Sort surge nuevamente del análisis estadístico de su comportamiento sobre distintos tipos de listas. La constante elegida como factor de reducción arroja para cualquier lista solamente tres combinaciones posibles para los últimos seis valores del gap: [9, 6, 4, 3, 2, 1], [10, 7, 5, 3, 2, 1] y [11, 8, 6, 4, 3, 2, 1]. Se ha determinado que se obtienen mejoras significativas en la velocidad de ejecución reemplazando por 11 el valor del gap cuando éste equivale a 9 o 10. Esta optimización es conocida como “CombSort11”. También existen



Para listas con cantidades bajas de elementos, Comb Sort tiene un rendimiento similar a Quick Sort. Ambos superan largamente a Bubble Sort.

“Otros investigadores establecieron que el factor de reducción más óptimo es la constante 1,247330950103979. Por razones prácticas, diversas implementaciones de Comb Sort utilizan redondeos de esta constante, como 1,25 o directamente 1,3.”



Cuando la lista es muy grande (millones de elementos) la performance de Comb Sort se degrada significativamente, mientras que la de Quick Sort apenas si se ve afectada.

versiones que en lugar de trabajar con un “factor de reducción”, recurren a una tabla predefinida para determinar el valor del gap en cada recorrida de la lista. Otra posibilidad, que también hemos planteado en otras oportunidades desde esta sección, es escribir un algoritmo de ordenamiento híbrido. Como sucede con otros métodos, inclusive algunos muy eficientes, como Quick Sort, mientras que en las primeras recorridas tiene lugar un número elevado de intercambios, en las últimas pasadas apenas si se hacen unos pocos, degradándose

significativamente su eficiencia. Una solución para este problema es trabajar con Comb Sort en las primeras recorridas, continuando el ordenamiento con otros sistemas, más efectivos al trabajar con listas “casi ordenadas”, como Insertion Sort, antes de que tengan lugar las últimas. Un buen momento para cambiar de estrategia al trabajar con listas largas es cuando el valor del gap es menor que 11.

ASPECTOS TEÓRICOS

La versión estándar de Comb Sort es mucho más eficiente que Bubble

Scripts.FranciscoCharrua.com

HotScripts.com - Over 10 web rings.
more links...

user name password
don't keep me logged ☐ log in
 Recommend this on Google

[Index](#) [News](#) [Samples](#) [E-Mail me](#) [Forums](#) [Sign Up](#)

SORT-COMPARE.JS

comb sort

-88	-46	-35	-33	-27	-21	-18	-16	-6	-5	-3	2	3	9	11	29	31	39	57	65	65	67	69	77	82
-----	-----	-----	-----	-----	-----	-----	-----	----	----	----	---	---	---	----	----	----	----	----	----	----	----	----	----	----

comparisons: 166
swaps: 43

quick sort

-88	-46	-35	-33	-27	-21	-18	-16	-6	-5	-3	2	3	9	11	29	31	39	57	65	65	67	69	77	82
-----	-----	-----	-----	-----	-----	-----	-----	----	----	----	---	---	---	----	----	----	----	----	----	----	----	----	----	----

comparisons: 84
swaps: 68

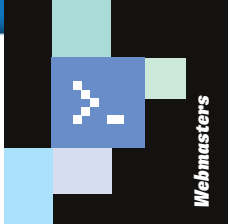
Sort. Para este último, la complejidad en el caso ideal es de $O(N)$, mientras que Comb Sort tiene la misma complejidad para el peor caso posible. Para el mejor caso y el caso promedio es de $O(N \log N)$, la misma que para Quick Sort, un algoritmo caracterizado por su eficiencia, aunque difieren en el peor caso, siendo para Quick Sort $O(N^2)$ y para Comb Sort $O(N \log N)$. Como pueden imaginar, con una lista de tan pocos elementos como la provista en los ejemplos aquí dados, no se notan tanto las diferencias entre estos tres métodos. Si planean ver con sus propios ojos las diferencias, les sugerimos que lo hagan empleando listas con cientos o miles de elementos. Una buena herramienta que los ayudará a crear listas “desordenadas” con grandes cantidades de valores es

la provista en el sitio web [<http://www.random.org/lists/>].

CONCLUSIONES

La palabra inglesa “comb” se traduce en español como “peine”. El nombre de este algoritmo proviene de aquellos peines que disponen de dos mitades, una en la cual los dientes se encuentran bastante separados entre sí, y que se utiliza en el comienzo del peinado, y otra donde los dientes están muy juntos y apretados, más apropiada para su empleo sobre el final de la operación. La analogía con la modalidad de trabajo del “gap”, que comienza comparando ambos extremos de la lista, para ir reduciéndose más y más tras cada iteración, queda así en evidencia. De esta forma concluimos una nueva vuelta de tuerca al popular Bubble Sort, que

nos deja una valiosa lección: no importa qué tan sencillo parezca un algoritmo, cuando creemos que lo tenemos totalmente aprendido y dominado, nos damos cuenta que todavía puede tener muchos aspectos que pueden mejorarse. El mismo criterio se aplica a todos los retos con los que nos enfrentamos los programadores en nuestra labor diaria: por más elegante y eficiente que parezca un bloque de código, siempre hay algo que puede optimizarse. **P**



Optimiza tu site con Google Webmaster Tools

Si alguien sabe de mejoras en la web es la marca del gigante navegador que por ser uno de los más utilizados se jacta de brindar la herramienta poderosa que le dará a tus páginas el empujoncito necesario para que funcionen al cien por ciento. En este artículo veremos unas recomendaciones para aprender a usarlo.

Por Roxana Miguel
roxana.miguel@pixelscode.com
@roxanamiguel



Webmaster es una de las profesiones de la era tecnológica que ya se ha ganado su lugar desde los '90 a esta parte. Están capacitados para generar soluciones en un sitio web ya que son los responsables del mantenimiento y la programación del mismo. Hasta hace un tiempo esta tarea era por demás conflictiva, ya que el desarrollo de las aplicaciones no daba tiempo a perfeccionarse con todas las técnicas. Atento a esto, Google pone sobre sus escritorios una herramienta llamada

Google Webmaster Tools, un aliado para simplificar el trabajo.

Aunque no es la solución para todos los problemas, al menos sí nos brinda los datos necesarios para comprobar el estado de indexación de las páginas en Internet y optimizar su visibilidad, que no es poco. Siempre que ofrecemos a nuestro cliente un sitio con todas las pautas que garanticen una mayor exposición, deberemos considerar esta aplicación que nos será muy útil no sólo durante el proceso de creación sino para solucionar pro-

blemas una vez que el site ya está en funcionamiento.

No se trata de un programa cuya instalación genera un sitio óptimo de por sí solo, sino que esta herramienta gratuita ofrecerá al webmaster un listado de datos que éste deberá saber interpretar para poder mejorar el site u orientarlo hacia el objetivo que tiene. En resumen estos son los beneficios que obtendremos:

- Comprobar el mapa del sitio.
- Conocer la frecuencia de indexación y las visitas de Googlebot.

- Obtener listados de enlaces que se brindan de nuestra web e incluso de los que nuestro site genera e interrumpen en la navegación.
- Comprobar las palabras claves que Google rastrea.
- Detectar errores en los links o enlaces.
- Determinar una URL preferida para que sea rastreada por los buscadores.

Esta herramienta no hace magia sino que, más bien, proporciona información valiosa para los Webmasters, debido a que con estos datos podremos solucionar errores en los enlaces, encontrar links rotos, desechar palabras claves inútiles para el buscador y utilizar otras más óptimas para los fines, entre otras particularidades. En pocas palabras es como tener una comunicación directa con Google ante cualquier inconveniente.

Nos será muy útil para mejorar la respuesta del sitio ante algún problema, con el control necesario sobre los resultados de búsqueda de Google y mejorar aspectos como la velocidad de carga o páginas no encontradas. Estos errores que nombrados azarosamente parecen equívocos de novatos, suelen ser muy comunes incluso hasta en los que mejor se manejan en la red.

¿CÓMO UTILIZAR GOOGLE WEBMASTER TOOLS?

Es muy fácil y, mejor aún, gratuito. Primero que nada crea una cuenta de Google que puede ser de Gmail o alguna de Google Apps; si ya la tienes, entonces vamos a poner las manos en la masa sin demoras. Deberás apuntar tu navegador a

<https://www.google.com/webmasters/tools/home?hl=es> y loguearte.



Esta herramienta no hace magia sino que, más bien, proporciona información valiosa para los Webmasters, debido a que con estos datos podremos solucionar errores en los enlaces, encontrar links rotos, desechar palabras claves inútiles para el buscador y utilizar otras más óptimas para los fines, entre otras particularidades."



Te encontrarás con un recuadro que te invita a colocar el nombre del sitio que has creado. Donde dice "Add a site" escribes el nombre de dominio y haces clic en "Continue", de esa forma estarás registrando tu sitio web.

Ahora será necesario verificar el site para lo cual podemos subir un archivo a la carpeta raíz del dominio, pudiendo ser por FTP que es lo más práctico. Una vez hecho esto presionamos el botón "Verify" que nos abrirá un menú de opciones en el caso de que la verificación haya sido positiva. Si esto no funcionara podemos

agregar una meta etiqueta en <head> con la home de nuestro site, o linkear nuestro perfil de Google Analytics con Google Webmaster Tools o, de lo contrario, incluir un registro en los DNS. Si nada de esto funciona, entonces el verificador ha logrado su cometido, descartar a quien quiere registrar un sitio del cual no es el Webmaster. Una suerte de valla de seguridad.

¿CUÁLES SON LAS HERRAMIENTAS DISPONIBLES?

Si hemos verificado correctamente el site, entonces podremos utilizar todas las herramientas que nos proporciona. Aquí veremos un recuento de las más importantes con una descripción para aprovecharlas al máximo:

- Mapa de sitio: Debemos enviar un sitemap en XML, de esta forma ayudamos al rastreo del GoogleBot, puesto que directamente le indicamos dónde está alojado nuestro XML. Cada vez que una persona busque información referida a nuestra página la encontrará más rápido, aun cuando lo que busque sea una página dentro del árbol del site, esto es porque la información que proporcionamos es un listado de todas las URL utilizadas. No hay que confundir esta herramienta con una aplicación para generar posicionamiento, esta no es la función, de hecho no tiene una función directa sino más bien el armado de un camino hacia nuestro site, que se puede dar o no, dependiendo más de otras razones que de la indexación del sitemap.

“Si bien Google Webmaster Tools no nos dice cómo solucionar nuestros errores en el web site, sí nos proporciona todos los datos necesarios para que con nuestros conocimientos podamos hacer una revisión rápida y mejorar la visibilidad de la página, así como también proporcionarles a los usuarios de nuestro sitio una buena experiencia de navegación.”



- Sitelinks: Cuando los usuarios googlean una palabra o frase aparecen todos los links relacionados a esta en el buscador. Los sitelinks son los enlaces extra que acompañan a un dominio principal. Se generan automáticamente y sirven para hacer que la experiencia de navegación de un usuario sea más dinámica, por esta razón no podemos producir nosotros mismos nuestros propios sitelinks, aunque sí podemos entrar al panel de Google Webmaster Central y en el menú Sitelinks ver todos los enlaces que se generaron y borrar los que no nos son útiles. Inclusive, suele sucedernos que finalmente hay enlaces

que no queremos mostrar y quedan predeterminados allí, pues será cuestión de acomodar los muebles de casa como queden mejor.

- Destinatarios: La opción orientación geográfica nos permite elegir hacia qué países queremos orientar el sitio que hemos creado. Aunque aprovechamos Internet para llegar a todo el mundo por igual y exponer una página hacia todos, nuestro cliente seguramente contará con un estudio de mercado o un mercado mismo interesado en sus productos; nosotros podemos potenciar un poco más esta particularidad por medio de esta herramienta.

Al entrar a la opción se muestra una leyenda que dice “Los usuarios a los que se orienta se encuentran en:”, colocando en el cuadro vacío junto a la frase el nombre del país, ya estamos avisándole a Google que este sitio debe tener más exposición en un lugar que en otro.

- Enlaces: Esta es una de las herramientas que los Webmasters con funciones SEO consultarán a menudo y es porque ofrece un listado de los sitios que enlazan nuestra página; nos brinda una información muy valiosa para saber desde qué sitios y hacia cuáles viran quienes nos contactan. Si bien tampoco sirve para posicionar la página, un buen SEO sabrá qué hacer con estos datos para lograr buenos resultados. Al entrar a la opción primero nos mostrará el total de links que enlazaron y abajo los enlaces y la cantidad de veces que cada uno de ellos entró.

- Errores: En la misma línea que la herramienta anterior, esta será otra buena opción para los SEO, ya que nos proporcionará un listado de URL's con errores. Tras el paso de GoogleBot, estos enlaces son como un bache en la ruta, pueden llevar nuestro sitio al fracaso, por lo que esta herramienta es una de las más importantes para prestar atención y solucionar cuanto antes. Google nos lo presenta como problemas que ha encontrado al rastrear nuestro sitio. Nos muestra un listado de enlaces con una descripción del error (por ejemplo, cuando no se encuentra el enlace el error es 404). A continuación se pueden ver desde qué cantidad de enlaces proviene el error y la fecha en que fueron detectados.

- Estadísticas: Cuando no encontramos ningún error rastreado, GoogleBot puede deslizarse por nuestro site sin problemas. Genera un reporte con las páginas que está indexando, los KB's que descargó nuestro sitio y el tiempo de descarga de los contenidos. Cuanta más información contamos en este reporte más visitas hemos obtenido de GoogleBot y eso nos asegura un buen posicionamiento con el tiempo.

- Sugerencias: Si tuviéramos la posibilidad de que un desarrollador de Google nos aconseje qué debemos revisar mirando nuestro sitio mientras lo estamos armando, digamos que esta herramienta es como si esto pudiera suceder. Nos proporciona un listado de consejos HTML, primero con una descripción general de la situación, si nuestro site tendrá inconveniente en los resultados de

búsqueda, si podemos mejorar el rendimiento y la experiencia de los usuarios o si todo está rotundamente mal confeccionado. A continuación nos detalla esos errores y en cuántas páginas figura el mismo error. Es importante estar atentos en caso de que nuestro sitio posea contenido no indexable, esto también lo advertirá la herramienta.

- Rendimiento: Este espacio está dedicado a evaluar cuánta dinámica ofrece nuestro sitio. Como se trata de un tema primordial para Google puesto que fija su atención en que la experiencia de los usuarios sea la apropiada, aquí los webmasters encontrarán suficiente tela para cortar. Nos proporciona una completa descripción sobre el rendimiento del sitio a fin de mejorar la velocidad y ofrecer un servicio más rápido para los usuarios. Para ello sabremos cuánto tarda en cargarse el sitio según la última actualización y en comparación con otros, es más, nos proporciona un gráfico referenciando los tiempos dividiendo en dos bandas: rápido y lento.

CONSEJOS PARA PRINCIPIANTES

Los webmasters que recién comienzan a usar estas herramientas o que a partir de este artículo se vean tentados a introducirse al mundo de Google Tools, además de experimentar las características de esta utilidad tendrán que tener en cuenta algunos consejos más que Google proporciona por medio de Webmaster Central Blog y que vendrán muy bien para todos los objetivos SEO:

Notificaciones por email: Configura Google Webmaster Tools para recibir información por email como alertas, evitando perder tiempo con la entrada y salida de la web para chequear los informes.

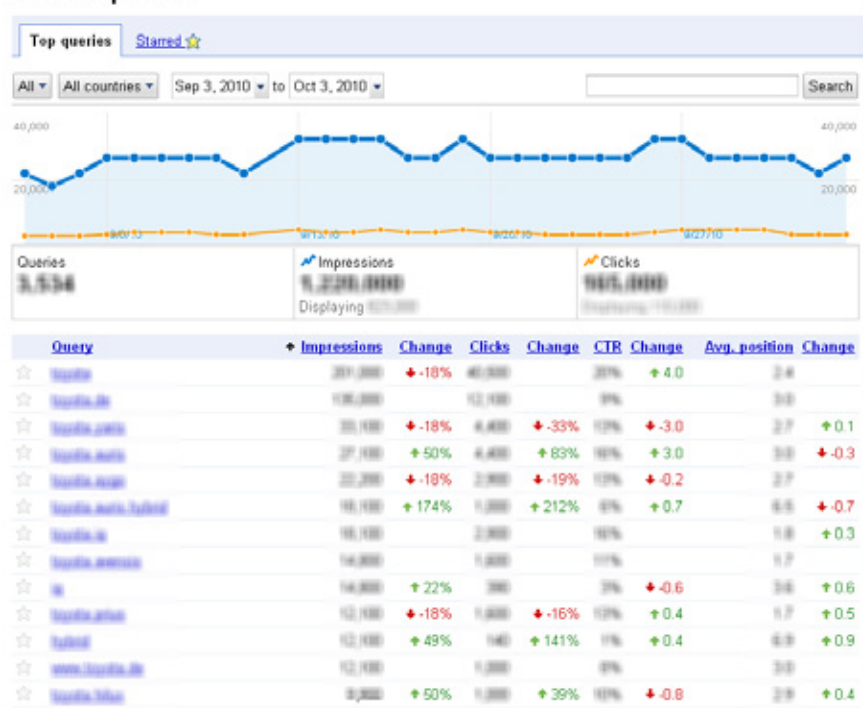
Organiza la revisión de datos: Consulta la búsqueda de datos actualizada con impresiones, posición media en página de resultados y más. No te quedes con datos desactualizados. Chequea cada cambio en la página con las herramientas para webmasters.

Palabras claves: Puedes mirar cuáles son las keywords que Google ha leído de nuestra web y ha marcado como relevantes. Estos datos nos servirán para optimizar nuestro sitio y plantearnos estrategias SEO a futuro.

Edad del sitio: Aunque muchos resalten la importancia de la antigüedad que nuestro site tiene para que GoogleBot lo capte, este es un dato no tan relevante. Bástate en tus informes actualizados que te asegurará más tiempo de vida.

Si bien Google Webmaster Tools no nos dice cómo solucionar nuestros errores en el web site, sí nos proporciona todos los datos necesarios para que con nuestros conocimientos podamos hacer una revisión rápida y mejorar la visibilidad de la página, así como también proporcionarles a los usuarios de nuestro sitio una buena experiencia de navegación. Esto es más que suficiente, si lo sabemos aprovechar, como para lograr una web exitosa. **P**

Search queries



Email Marketing con tu marca

Ganá dinero ofreciendo a tus
clientes la **mejor** herramienta.



EnvialoSencillo.com

dattatec.com