novática
CEPIS UPGRADE
Revista de la Asociación de Técnicos de Informática

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). ***Novática*** edita asimismo **UP**GRADE, revista digital de **CEPIS** (*Council of European Professional Informatics Societies*), en lengua inglesa, y es miembro fundador de **UP**ENET (**UP**GRADE ***E****uropean* ***NET****work*).

<http://www.ati.es/novatica/>
<http://www.ati.es/reicis/>
<http://www.upgrade-cepis.org/>

ATI es miembro fundador de **CEPIS** (*Council of European Professional Informatics Societies*) y es representante de España en **IFIP** (*International Federation for Information Processing*); tiene un acuerdo de colaboración con **ACM** (*Association for Computing Machinery*), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AI2**, **ASTIC**, **RITSI** e **Hispalinux**, junto a la que participa en **ProInnova**.

Consejo Editorial
Antoni Carbonell Nogueras, Juan ManuelCueva Lovelle, Juan Antonio Esteban Iriarte,Francisco López Crespo, Julián Marcelo Cocho, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codina, Fernando Piera Gómez (Presidente del Consejo), Ramón Puigjaner Trepat, Miquel Sárries Griñó, Asunción Yturbe Herranz

Coordinación Editorial
Llorenç Pagés Casas <pages@ati.es>
Composición y autoedición
Jorge Llácer Gil de Ramales
Traducciones
Grupo de Lengua e Informática de ATI <http://www.ati.es/gt/lengua-informatica/>, Dpto. de Sistemas Informáticos - Escuela Superior Politécnica - Universidad Europea de Madrid
Administración
Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas - Coordinadores
Acceso y recuperación de la información
José María Gómez Hidalgo (Optenet), <jmgomezh@yahoo.es>
Manuel J. Maña López (Universidad de Huelva), <manuel.mana@diesia.uhu.es>
Administración Pública electrónica
Francisco López Crespo (MAE), <flc@ati.es>
Arquitecturas
Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>
Jordi Tubella Murgadas (DAC-UPC), <jordit@ac.upc.es>
Auditoría SITIC
Marina Touriño Troitiño, <marinatourino@marinatourino.com>
Manuel Palao García-Suelto (ASIA), <manuel@palao.com>
Derecho y tecnologías
Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV), <ihernando@legaltek.net>
Elena Davara Fernández de Marcos (Davara & Davara), <edavara@davara.com>
Enseñanza Universitaria de la Informática
Joaquín Ezpeleta Mateo (CPS-UZAR), <ezpeleta@posta.unizar.es>
Cristóbal Pareja Flores (DSIP-UCM), <cpareja@sip.ucm.es>
Entorno digital personal
Alonso Alvarez García (TID), <aag@tid.es>
Diego Gachet Páez (Universidad Europea de Madrid), <gachet@uem.es>
Estándares Web
Encarna Quesada Ruiz (Oficina Española del W3C) <equesada@w3.org>
José Carlos del Arco Prieto (TCP Sistemas e Ingeniería) <jcarco@gmail.com>
Gestión del Conocimiento
Joan Baiget Solé (Cap Gemini Ernst & Young), <joan.baiget@ati.es>
Informática y Filosofía
José Angel Olivas Varela (Escuela Superior de Informática, UCLM) <joseangel.olivas@uclm.es>
Karim Gherab Martín (Harvard University) <kgherab@gmail.com>
Informática Gráfica
Miguel Chover Sellés (Universitat Jaume I de Castellón), <chover@lsi.uji.es>
Roberto Vivó Hernando (Eurographics, sección española), <rvivo@dsic.upv.es>
Ingeniería del Software
Javier Dolado Cosín (DLSI-UPV), <dolado@si.ehu.es>
Luis Fernández Sanz (PRIS-EI-UEM), <lufern@dpris.esi.uem.es>
Inteligencia Artificial
Vicente Botti Navarro, Vicente Julián Inglada (DSIC-UPV)
<{vbotti, vinglada}@dsic.upv.es>
Interacción Persona-Computador
Julio Abascal González (FI-UPV), <julio@si.ehu.es>
Lengua e Informática
M. del Carmen Ugarte García (IBM), <cugarte@ati.es>
Lenguajes Informáticos
Andrés Marín López (Univ. Carlos III), <amarin@it.uc3m.es>
J. Angel Velázquez Iturbide (ESCET-URJC), <a.velazquez@escet.urjc.es>
Lingüística computacional
Xavier Gómez Guinovart (Univ. de Vigo), <xgg@uvigo.es>
Manuel Palomar (Univ. de Alicante), <mpalomar@dlsi.ua.es>
Mundo estudiantil y jóvenes profesionales
Federico G. Mon Trotti (RITSI) <gnu.fede@gmail.com>
Mikel Salazar Peña (Area de Jovenes Profesionales, Junta de ATI Madrid), <mikeltxo_uni@yahoo.es>
Profesión Informática
Rafael Fernández Calvo (ATI), <rfcalvo@ati.es>
Miquel Sárries Griñó (Ayto. de Barcelona), <msarries@ati.es>
Redes y servicios telemáticos
José Luis Marzo Lázaro (Univ. de Girona), <joseluis.marzo@udg.es>
Germán Santos Boada (UPC), <german@ac.upc.es>
Seguridad
Javier Areitio Bertolín (Univ. de Deusto), <jareitio@eside.deusto.es>
Javier López Muñoz (ETSI Informática-UMA), <jlm@lcc.uma.es>
Sistemas de Tiempo Real
Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM),
<{aalonso,jpuente}@dit.upm.es>
Software Libre
Jesús M. González Barahona, Pedro de las Heras Quirós (GSYC-URJC),
<{jgb,pheras}@gsyc.escet.urjc.es>
Tecnología de Objetos
Jesús García Molina (DIS-UM), <jmolina@um.es>
Gustavo Rossi (LIFIA-UNLP, Argentina), <gustavo@sol.info.unlp.edu.ar>
Tecnologías para la Educación
Juan Manuel Dodero Beardo (UC3M), <dodero@inf.uc3m.es>
César Pablo Córcoles Briongo (UOC), <ccorcoles@uoc.edu>.
Tecnologías y Empresa
Didac López Viñas (Universitat de Girona), <didac.lopez@ati.es>
Francisco Javier Cantais Sánchez (Indra Sistemas), <fjcantais@gmail.com>
TIC y Turismo
Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga)
<{aguayo, guevara}@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de losmismos. ***Novática*** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o *copyright* elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a ***Novática*** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Padilla 66, 3º, dcha., 28006 Madrid
Tlfn.914029391; fax.913093685 <novatica@ati.es>
Composición, Edición y Redacción ATI Valencia
Av. del Reino de Valencia 23, 46005 Valencia
Tlfn./fax 963330992 <secreval@ati.es>
Administración y Redacción ATI Cataluña
Vía Laietana 46, ppal. 1ª, 08003 Barcelona
Tlfn.934125235; fax 934127713 <secregen@ati.es>
Redacción ATI Andalucía
Isaac Newton, s/n, Ed. Sadiel,
Isla Cartuja 41092 Sevilla, Tlfn./fax 954460779 <secreand@ati.es>
Redacción ATI Aragón
Lagasca 9, 3-B, 50006 Zaragoza.
Tlfn./fax 976235181 <secreara@ati.es>
Redacción ATI Asturias-Cantabria <gp-astucant@ati.es>
Redacción ATI Castilla-La Mancha <gp-clmancha@ati.es>
Suscripción y Ventas <http://www.ati.es/novatica/interes.html>, ATI Cataluña, ATI Madrid
Publicidad
Padilla 66, 3º, dcha., 28006 Madrid
Tlnf.914029391; fax.913093685 <novatica@ati.es>
Imprenta: Derra S.A., Juan de Austria 66, 08005 Barcelona.
Depósito legal: B 15.154-1975 – ISSN: 0211-2124; CODEN NOVAEC
Portada: "Salida de la habitación 101" - Concha Arias Pérez / © ATI
Diseño: Fernando Agresta / © ATI 2003

Nº 190, noviembre-diciembre 2007, año XXXIII

sumario

editorial
Estudiantes y jóvenes profesionales, clave del futuro de ATI > 02

en resumen
El poder de laas comunidades > 02
Llorenç Pagés Casas

monografía

Sortware libre: investigación y desarrollo
(En colaboración con UPGRADE*)*
Editores invitados: *Manuel Palomo Duarte, José Rafael Rodríguez Galván, Israel Herraiz Tabernero y Andrea Capiluppi*

Presentación. Sofware libre: investigación y desarrollo > 03
Andrea Capiluppi, José Rafael Rodríguez Galván, Manuel Palomo Duarte, Israel Herraiz Tabernero

La necesidad de investigar sobre software libre en Europa > 06
Israel Herraiz Tabernero, Rafael Rodríguez Galván, Manuel Palomo Duarte

De la catedral al bazar: un estudio empírico del ciclo de vida de los proyectos basados en comunidades de voluntarios > 09
Andrea Capiluppi, Martin Michlmayr

Los bienes comunes como nueva economía y lo que esto significa para la investigación > 17
Richard P. Gabriel

Software libre para la gestión de proyectos de investigación > 20
Israel Herraiz Tabernero, Juan José Amor Iglesias, Álvaro del Castillo San Félix

Innovación tecnológica en comunicaciones móviles desarrollada con Software Libre: Campus Ubicuo > 25
Javier Carmona Murillo, José Luis González Sánchez, Manuel Castro Ruiz

El modelo de la Oficina de Software Libre de la Universidad de Cádiz en la universidad española > 31
José Rafael Rodríguez Galván, Manuel Palomo Duarte, Juan Carlos González Cerezo, Gerardo Aburruzaga García, Antonio García Domínguez, Alejandro Álvarez Ayllón

Aprendiendo a introducir una innovación en un proyecto basado en Software Libre > 36
Christopher Oezbek, Lutz Prechelt

Optimización del proceso de *render* 3D distribuido con software libre > 41
Carlos González Morcillo, Gerhard Weiss, David Vallejo Fernández, Luis Jiménez Linares, Javier Albusac Jiménez

secciones técnicas

Mundo estudiantil y jóvenes profesionales

SWAML, Semantic Web Archive of Mailing Lists > 49
Sergio Fernández López, Diego Berrueta Muñoz, José Emilio Labra Gayo

TCOS: uso de terminales ligeros en las aulas > 52
Mario Izquierdo Rodríguez

Porting de GCC al microcontrolador Microchip PIC16F877 > 55
Pedro José Ramírez Gutiérrez

SubDownloader > 58
Iván García Cortijo

Software Libre en la Enseñanza: primeras jornadas organizadas por OuSLi en el ámbito de la educación > 61
José Ramón Méndez Reboredo, Enrique Estévez Fernández, Florentino Fernández Riverola, Daniel González Peña

Referencias autorizadas > 64

sociedad de la información

Nueva Economía

Las TIC y la Ciencia, Ingeniería y Gestión de los Servicios > 69
Gregorio Martín Quetglas, Vicente Cerverón Lleó, Francisco J. Gálvez Ramírez

Programar es crear

Todas las palabras son capicúas (CUPCAM 2006, problema F, solución) > 73
Óscar Martín Sánchez

Las luces de la escalera (CUPCAM 2006, problema G, enunciado) > 74
Julio Mariño Carballo

Permutaciones con un número dado de inversiones (CUPCAM 2006, problema H, enunciado) > 75
Manuel Abellanas Oar, Luis Hernández Yáñez

asuntos interiores

Coordinación Editorial / Programación de Novática > 76
Normas para autores / Socios Institucionales > 77

Monografía del próximo número: "*Gobierno de las TIC*"

Óscar Martín Sánchez
Universidad Complutense de Madrid

<omartin@estad.ucm.es>

Todas las palabras son capicúas

El enunciado de este problema apareció en el número 189 de ***Novática*** (septiembre-octubre 2007, p. 72). Es el problema F de los planteados en el IV Concurso Universitario de la Comunidad Autónoma de Madrid (CUPCAM 2006) del que ATI fue entidad colaboradora.

Recordemos que el problema consiste en dividir una cadena de caracteres en trozos que la hagan capicúa por trozos, como en `ca-rra-ca`.

Una idea que viene fácilmente a la cabeza consiste en recorrer la cadena simultáneamente desde los dos extremos, comparando a cada paso el prefijo y el sufijo que tenemos delimitados: si no coinciden, seguimos hacia dentro; si sí son iguales, insertamos los guiones de separación y seguimos trabajando con la parte central pendiente. Este método es eficiente y sencillo de implementar.

En este artículo, sin embargo, describiremos otro método, menos evidente y más interesante, que usa expresiones regulares. La expresión regular que está en la base de la solución es ésta:

`(.+)(.*)\1`

Recuérdese que los paréntesis sirven para delimitar subcadenas de la cadena de caracteres que se ajuste a la expresión regular; y `\1` es la forma de referirnos a la primera de esas subcadenas desde dentro de la propia expresión regular. Por tanto, una cadena que se ajuste a la expresión regular de arriba debe tener un prefijo no vacío cualquiera (`.+`), un sufijo exactamente igual que el prefijo (`\1`), y una parte intermedia cualquiera, que incluso puede ser vacía (`.*`).

La utilidad de esa expresión regular en nuestro problema empieza a vislumbrarse, esperamos. Necesitamos, sin embargo, un pequeño retoque. Considérese la cadena `AABBAA`; se puede ajustar a nuestra expresión regular de dos maneras distintas: bien `A-ABBA-A` o bien `AA-BB-AA`. Es decir, podemos escoger para el primer grupo sólo una A o dos, y ambas cosas producen ajustes válidos. Como nosotros queremos dividir las cadenas en trozos lo más pequeños posible, lo que nos interesa en este caso es tomar sólo una `A`. Sin embargo, los motores de ajuste de expresiones regulares hacen la otra elección, con el primer grupo lo más largo posible. Para cambiar ese comportamiento necesitamos añadir un símbolo a nuestra expresión regular:

`(.+?)(.*)\1`

Ese interrogante modifica el significado del cuantificador `+`, haciéndolo perezoso. Basándonos en esta expresión regular, vamos a realizar sustituciones en la cadena de entrada, dentro de un bucle. Para que las sustituciones sólo afecten a la parte de cadena que aún no haya sido estudiada y dividida, necesitamos introducir unos delimitadores. Elegimos para ello los caracteres `<` y `>`. (La forma en la que los vamos a usar implica que si la cadena de entrada ya contiene alguno de estos caracteres, nuestro algoritmo no funcionará. Nos tomamos, pues, la licencia, que no forma parte del enunciado del problema, de suponer que esos caracteres no están en ninguna de las cadenas de entrada.) La sustitución que proponemos es

cambiar `<(.+?)(.*)\1>` por `$1-<$2>-$1`

La sintaxis `$1`, `$2`, etc, es habitual para referirse, desde fuera de la expresión regular, a los grupos marcados por los paréntesis, es decir, a lo mismo a lo que `\1`, `\2`, etc. se refiere desde dentro de la expresión regular. Explicada en palabras, esa sustitución sería algo así: Si la parte de cadena delimitada por `<` y `>` se puede dividir en tres trozos según se explicó arriba, sáquense de los delimitadores el primer trozo y el tercero, insértense los guiones separadores, y déjese el grupo intermedio como nueva cadena a estudiar. Esta sustitución deberá hacerse tantas veces como sea posible.

Veamos las sustituciones en acción sobre nuestra cadena de ejemplo, `AABBAA`, una vez encerrada entre los delimitadores:

<AABBAA>
A-<ABBA>-A
A-A-<BB>-A-A
A-A-B-<>-B-A-A

Evidentemente, hacen falta algunos arreglos para obtener el resultado final: hemos de quitar los delimitadores y hemos de sustituir los dos guiones que quedarán en el centro por uno solo (esos dos guiones seguidos sólo aparecen cuando la cantidad final de trozos es par).

Eso es todo. Estas ideas quedan plasmadas, por ejemplo, en el programa Java:

```
import java.io.*;
public class palindr
{
  public static void main(String arg[]) throws
IOException
  {
    BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
    String s, aux;
    s = in.readLine();
    while (s!=null)
    {
       aux = "<" + s + ">";
       while (!aux.equals(s))
       {
          s = aux;
          aux =s.replaceFirst("<(.+?)(.*)\\1>","$1-
<$2>-$1");
       }
       s = s.replaceFirst("<","").
        replaceFirst(">","").replaceFirst("—","-");
       System.out.println(s);
       s = in.readLine();
    }
  }
}
```

El método `replaceFirst`, de la clase `String`, hace exactamente lo que necesitamos. Nótese que usamos ese mismo método para los toques finales (eliminación de los delimitadores y del guión repetido). Nótese también la condición de final de bucle: paramos cuando la cadena resultado de la sustitución sea igual que la anterior, es decir, cuando no se haya podido hacer tal sustitución (porque la cadena entre los delimitadores no se haya ajustado a la expresión regular). Esa condición es seguramente mejorable, pero tal como está es sencilla y suficiente para nuestros fines. (Continúa en página siguiente).