

Los proyectos de los nuevos desarrolladores españoles

3ER. CONCURSO UNIVERSITARIO SOFTWARE LIBRE

Publicamos una serie de mini-artículos de los proyectos presentados a la Tercera Edición del Concurso Universitario de Software Libre escritos por los participantes.

ArCan

En estos últimos años, Arduino [1] un sistema de desarrollo basado en AtMega8/16, ha propulsado el HW libre y ha creado un movimiento que, de no variar el rumbo, será dentro de poco tan importante como ha conseguido serlo el SW libre. Son ya muchos los que de forma profesional o personal se dedican a compartir sus avances con el resto de la comunidad, enriqueciendo sin duda a todo aquel que de una forma u otra necesite hacer uso de estos conocimientos.

¿Por qué Nace ArCan?

Almorzando con uno de mis antiguos profesores de la Universidad de Málaga, David Santo Orcero, salió la conversación sobre domótica y HW libre, y me propuso hacer un Shield para Arduino que permitiese intercomunicarlos vía CAN-Bus (Figura 1) como proyecto fin de carrera, y presentarme con dicho proyecto al Concurso Universitario de Software Libre. ¡Una idea que no pude rechazar!

Los objetivos

Inicialmente el proyecto sólo se limitaba a crear el Shield y las librerías necesarias para usarlo con Arduino. Sin embargo, durante su desarrollo decidí implicarme aún más; buscando no sólo

un PFC para cumplir el trámite, sino algo que fuera de utilidad práctica para la comunidad, y para que pudiesen hacerlo en sus casas los que no estén acostumbrados a trabajar con hardware.

Por ello, ArCan, cuyo blog se encuentra en [2] no tiene como único objetivo la creación del shield, sino también mostrar cómo usar el bus de comunicación SPI, entender las grandes ventajas que aporta CAN frente a otros sistemas, manejo de los datasheets, etc. Un proyecto de estas características tiene una gran carga técnica, pero en la medida de lo posible simplifico las explicaciones para que sean fáciles de entender.

¿Por qué CAN?

CAN, Controller Area Network, que nace inicialmente como respuesta al incremento de dispositivos electrónicos en el automóvil, aporta una gran cantidad de ventajas en escenarios donde la fiabilidad en las comunicaciones prima frente a otros factores. CAN sigue una topología Cliente-Servidor, que es la más usada en el ámbito industrial, cuenta con mecanismos propios de detección de errores, usa un bus serie, con la ventaja del ahorro de materia prima y además es multimaestro, todos estos factores con otros muchos que se me hace imposible

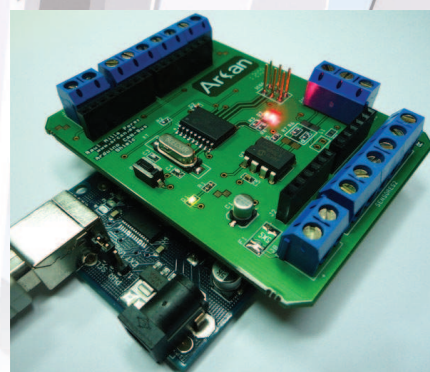


Figura 1: Arduino junto con el Shield ArCan.

relatar en 500 palabras, hacen de CAN un protocolo robusto, fiable y seguro.

El futuro de ArCan

Una vez creado el shield, el siguiente paso será crear comunidad alrededor del proyecto. Como no todo el mundo dispone de soldador ni tiene experiencia soldando en superficie, voy a fabricar una pequeña tirada que se pondrá a la venta en el blog, sin ánimo de lucro, intentando que la unidad esté por debajo de los 30€.

Agradecimientos

Quiero terminar el artículo agradeciendo la ayuda de David, mi director de proyecto; de Ingelabs [3], empresa en la que trabajo y que ha patrocinado los costes del prototipado y por último, a Ana

Cruz Martín, una antigua profesora que siempre está cuando la necesito. Sin ellos llegar hasta aquí hubiese sido mucho más difícil.

Autor

Raúl Milla Pérez

RECURSOS

- [1] Proyecto Arduino: <http://www.arduino.cc>
- [2] Blog del proyecto ArCan: <http://www.arcan.es>
- [3] Ingelabs: <http://www.ingelabs.com>

Blue Harvest

El proyecto Blue Harvest nace con el propósito de facilitar, de una forma muy sencilla y amena, las reservas de aulas y de materiales que se realizan en la Escuela Universitaria Politécnica de Cuenca (E.U.P.C.). En un primer momento éste era el objetivo, aunque en las primeras fases de planificación del proyecto decidimos ampliar las miras de la aplicación, y no sólo limitarla únicamente al edificio de la E.U.P.C., sino que fuera útil para cualquier edificio de la Universidad de Castilla-La Mancha u otro centro.

Blue Harvest es una aplicación web basada en php que facilita a los usuarios la reserva de dependencias, el equipamiento necesario para reuniones, clases, etc., además de la fecha y hora en las que van a tener lugar.

Motivaciones

Nuestra principal motivación era participar en el desarrollo de una aplicación informática, ya que la experiencia que poseíamos en este campo era nula. Por otro lado, pretendíamos ahondar en otros lenguajes de programación, ya que en nuestra especialidad, Sonido e Imagen, no se profundiza mucho en la programación.

Otras motivaciones que teníamos eran las del trabajo bien hecho, el esfuerzo y

el compañerismo. Partiendo de estas motivaciones básicas nacieron otras, como la de ayudar a la difusión del Software Libre entre la gente que nos conoce, y enseñarle que este “mundo” no es tan complicado como ellos lo ven.

Desarrollo de la Aplicación

Los primeros pasos que seguimos en este largo camino del desarrollo de cualquier aplicación fueron reuniones con todos los usuarios potenciales de la aplicación, profesores y personal de la escuela. De ellos conseguimos nuevas ideas y descartamos cosas que pensábamos no consideraban importantes.

El siguiente paso fue la definición de las tablas que utilizaríamos en la base de datos, siendo este uno de los puntos fundamentales para comenzar con buen pie todo y no tener que estar haciendo cambios en mitad del proyecto.

A partir de aquí comenzamos a desarrollar todo el entorno web. Se crearon tres entornos diferentes, uno por cada usuario. Cada usuario posee herramientas específicas: para el Administrador se creó todo lo necesario para la gestión de Blue Harvest; el perfil de usuario tiene cuanto necesita para crear las reservas, además de conocer en todo momento el estado de sus reservas. Por último, el personal del centro puede ver todas las reservas, lo que le permite hacer todos los preparativos para el día y hora fijados en la reserva.

Conclusión

Desde nuestro punto de vista, hemos conseguido cumplir con nuestros objetivos iniciales, desarrollar una aplicación sencilla y vivir una experiencia nueva. Por todo esto creo que nuestro esfuerzo y trabajo han merecido la pena.

Autores

Eduardo Villar Rabadán y Alejandro Olmedo Mateo. Estudiantes de Ingeniería Técnica de

Cool Imaging

Un sistema de caracterización global y local de imágenes digitales mediante extracción de rasgos basados en contenido.

Motivación

Nuestro objetivo es el de contribuir al desarrollo científico a través de un proyecto de caracterización de imágenes mediante la extracción de rasgos basados en contenido, donde se puedan realizar estudios globales y locales. El sistema

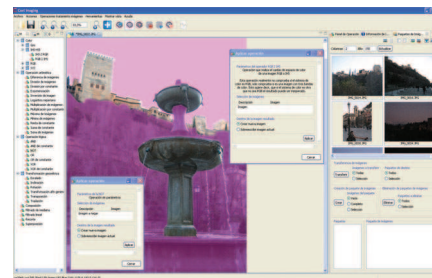


Figura 3: Captura de la aplicación donde se ha seleccionado una región de interés (ROI) y se tienen abierta dos ventanas de operaciones.

permite, en última instancia, caracterizar tanto global como localmente una imagen digital mediante rasgos matemáticos de utilidad para la propia caracterización. La idea que nos mueve es la de crear una herramienta de propósito general que aglutine una gran diversidad de técnicas de caracterización basadas en contenido, que pueda ser usada por todo tipo de especialistas, independientemente del campo en el que se muevan. Al tratarse de una aplicación de tratamiento de imágenes, ésta debe incluir herramientas clásicas de manipulación de imágenes digitales, independientemente de la caracterización que posteriormente se pudiera hacer de éstas.

La caracterización de imágenes es un campo en continuo avance. El avance científico en este área permite el desarrollo de nuevas técnicas de caracterización. Es por ello que el proyecto se ha desarrollado con la filosofía de *ampliabilidad* en mente: una estructura fácilmente extensible permitirá la incorporación de nuevas técnicas y algoritmos para los que no se haya dado soporte inicialmente.

Desarrollo y Funcionamiento

Teniendo en cuenta el concepto de ampliabilidad y que la aplicación pueda ser utilizada por cualquier tipo de especialista, el desarrollo del proyecto se ha visto altamente influenciado por la herramienta de desarrollo seleccionada. Se eligió el framework *Eclipse Rich Client Platform* por dos razones fundamentales. Por un lado, permite aprovechar la potencia del framework Eclipse para nuestra aplicación, especialmente el sistema de Plug-in's, de manera que una persona con conocimientos de programación pueda implementar nuevas operaciones sin tener que conocer el funcionamiento interno de la aplicación.



Figura 2: Vista de usuario de BlueHarvest.

Por otro lado, la estética homogénea que provee, que junto al diseño realizado, trata que la utilización de la aplicación sea intuitiva y homogénea, haciendo que las operaciones se puedan realizar a golpe de click e introduciendo los parámetros requeridos por las operaciones. Una operación en la aplicación se llevaría a cabo de la siguiente manera: se abre una imagen, se dirige uno al menú izquierdo de operaciones, selecciona la operación deseada, introduce los parámetros si la operación los requiere y aplica la operación sobre la imagen haciendo click en aplicar.

Autores

Luis A. González Jaime y Ricardo J. Palma Durán, estudiantes de Ingeniería Informática de la Universidad de Granada.

RECURSOS

- [1] Más información: <https://forja.rediris.es/projects/cusl3-coolimage>

eOPSOA, calidad del software para vagos

Es común que los sectores críticos al Software Libre postulen que éste es de menor calidad que su contrapartida propietaria. Sin desmerecer a proyectos de excelente calidad, es verdad que la atención a la calidad es una de las asignaturas pendientes en el mundo del Software Libre. Realizar pruebas es aburrido, se pierde tiempo y es poco gratificante, además de ser inherentemente difícil.

Mi proyecto en el III CUSL se llama eOPSOA, y es una herramienta que permitirá a los proyectos de Software Libre mejorar gracias a la metodología de calidad OPSOA (Open SOURCE Assesment). Creada por los profesores de la UCLM Elena Navarro y Francisco Montero, tiene como objetivo certificar que un programa cumple una serie de requisitos de usabilidad, funcionalidad y mantenibilidad. eOPSOA permitirá a los proyectos analizar su aplicación, ahorrando una cantidad ingente de tiempo y esfuerzo. La Figura 1 muestra una descripción del proceso que sigue OPSOA.

eOPSOA es una herramienta ambiciosa porque examina varios aspectos fundamentales de la calidad del software. El estudio de la *usabilidad* se hace comple-

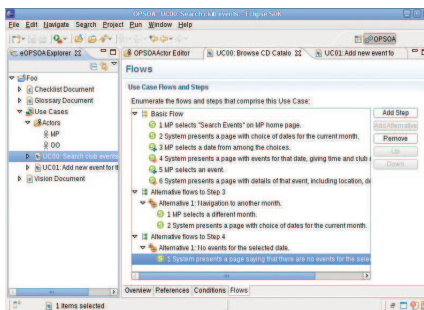


Figura 4: Descripción de los pasos que componen un Caso de Uso en eOPSOA.

tando una serie de cuestionarios que expondrán características de interés del programa. La *descripción funcional* se realiza mediante la especificación de los Casos de Uso del programa (véase la Figura 2). Respecto a la *mantenibilidad* y *seguridad*, se integrarán herramientas de análisis estático del código, como pueden ser CCCC, RATS o JavaNCSS.

Quien haya testeado alguna vez la funcionalidad de una aplicación, sabrá que no es una tarea trivial. eOPSOA será de gran ayuda para los testers porque les permitirá organizar de forma ágil sus tests de pruebas, archivar sus resultados y estudiar la evolución de los programas sujetos a estudio. Pero eOPSOA no se queda ahí, sino que en un futuro, y a partir de la definición de los Requisitos Funcionales de un programa (véase la Figura 3), podrá generar de forma automática una propuesta de conjuntos de tests de pruebas que sin lugar a dudas, ahorrará una cantidad brutal de trabajo. Se está estudiando integrar herramientas de testeo GUI, como Dogtail de Red Hat, para hacer aún más fácil el análisis funcional de las aplicaciones.

Una vez finalizado el análisis, eOPSOA mostrará una puntuación y una serie de consejos que los desarrolladores podrán utilizar para mejorar la calidad de sus aplicaciones.

eOPSOA está programada en Java, se ejecuta sobre OpenJDK 6 y se distribuye como un plugin para Eclipse. Si quieres saber más, tenemos un blog donde se documenta el proceso de desarrollo y una forja donde está alojado el repositorio del proyecto. Si te parece interesante mi proyecto y quieres echar una mano, ¡bienvenido!, te espero en la forja :-)

Autor

David Castellanos Serrano. Estudiante de Ingeniería Informática en la Universidad de Castilla-La Mancha.

RECURSOS

- [1] Blog: <http://eopsoa.blogspot.com>
 [2] Forja: <http://forja.molinux.info/projects/eopsoa>
 [3] Update Site: <http://eopsoa.albacete.org>
 [4] OpenJDK: <http://openjdk.java.net/>

GECO

GECO pretende ser un gestor de contraseñas y ficheros de configuración distribuido con diferentes interfaces de comunicación con el usuario. Presenta tanto una interfaz de escritorio en GTK como una interfaz web desarrollada con python y webpy.

GECO es un proyecto presentado a la tercera edición del Concurso Universitario de Software Libre que nace de la necesidad de tener contraseñas aleatorias y diferentes para cada servicio web (o no) en el cual estás registrado.

Problema

Hoy en día la mayoría de la gente que navega con regularidad tiene multitud de cuentas creadas en diferentes páginas webs; estas cuentas, normalmente, están basadas en usuario y contraseña. Las contraseñas se almacenan de diferentes formas en las bases de datos de los diferentes servicios, de tal forma que no podemos saber si el administrador o un intruso pueden llegar a ver nuestra contraseña.

Por tanto, es realmente importante utilizar claves diferentes (y aleatorias) para cada aplicación, pero memorizar 5, 10 ó 20 claves seguras es realmente complicado, y para esto están los gestores de contraseñas.

Sin embargo, el uso de un gestor implica que, si no recuerdas tus contraseñas, estás perdiendo la movilidad y la libertad de usar diferentes máquinas en las cuales no tengas instalado el gestor con tus claves personales. Además, en el caso de rotura de tu disco duro, puedes llegar a perder la base de datos, perdiendo así para siempre muchas contraseñas que no podrás recuperar.

Solución

GECO pretende dar solución a los problemas que presentan los gestores de contraseñas de escritorio, yendo un paso más allá y prestando un servicio de almacenamiento y gestión de distribuidos (o centralizados) con dife-

Concurso Universitario de Software Libre



Libre

Fase Final
7 y 8 de Mayo 2009

<https://www.concursosoftwarelibre.org>

Patrocinador Principal



Patrocinador Oro



Patrocinador Plata



Patrocinador Bronce



Organiza



Medios Oficiales



Colaborador Principal



Universidades Colaboradoras



UNIVERSIDAD DE SALAMANCA

Organismos Colaboradores



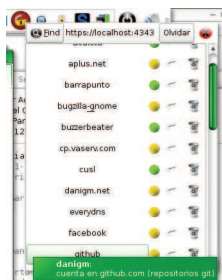


Figura 5: El cliente gtk es un icono de sistema desde el cual se copia la contraseña al portapapeles y no es necesario ni siquiera verla.

por tanto toda información que salga del cliente irá cifrada con una clave maestra, y de esta manera en el servidor no se almacenará información recuperable sin esta contraseña.

Arquitectura

GECO implementa una arquitectura clásica cliente-servidor, de tal forma que hay un servidor GECO al cual se pueden conectar diferentes clientes para administrar contraseñas.

De este modo, un usuario puede tener un servidor GECO en su máquina y utilizarlo normalmente como un gestor de contraseñas normal, pero además puede sincronizar su servidor con otro en una máquina remota, de tal forma que si el usuario se mueve, pueda tirar de la interfaz web del servidor de la máquina remota para administrar o acceder a sus contraseñas.

¿Algo más?

Además de todo esto, GECO ofrecerá la posibilidad de almacenar ficheros de configuración (todo cifrado), de manera que en un linux con un cliente GECO puedas utilizar tu fichero de configuración de Pidgin sólo durante la sesión y posteriormente se eliminará todo rastro de tus ficheros de configuración, quedando el sistema tal y como estaba.

Licencia y otros Asuntos

Todo lo que se desarrolle será bajo licencia GPLv3 y todo el artwork y documentación será totalmente libre con la licencia “úsalo como quieras y para lo que quieras”.

La idea de este proyecto es hacerme un gestor de contraseñas para mí, y si le sirve a otra persona pues mejor.

Autor

Daniel García Moreno.

RECURSOS

[1] Web del proyecto: <http://danigm.net/geco>

Gesport

Gesport es un programa que permite la planificación deportiva según la metodología que siguen los entrenadores. Sirve como herramienta de comunicación con el deportista, que aporta un feedback, y permite el análisis del cumplimiento de la planificación.

El hacer este proyecto no ha sido precisamente una elección, sino algo que esperaba y al final llamó a mi puerta. Tras hacer un primer intento de proyecto de fin de carrera sobre casas rurales, pensé que la mejor manera de acabar mis estudios era con algo realmente apasionante y que me llenase. Después de 20 años haciendo atletismo el proyecto tenía que versar sobre deporte y, si podía ser, tener algo que ver con la informática. El software propietario nunca fue una opción.

La matemática en el estudio del cuerpo

Para entender porqué es necesario un programa informático para el estudio del rendimiento nos podemos quedar en los datos numéricos que usa la biomecánica actual, que trata el cuerpo humano como ciencia. A la hora de la verdad cualquier entrenador se distingue de los demás por su aporte personal, que hace sus entrenamientos únicos, casi artísticos.

No es casualidad que los primeros seres humanos en representar el cuerpo a tamaño natural y de forma proporcionada fueran los griegos. Ellos fueron los creadores de los primeros Juegos Olímpicos y tenían verdadero culto al cuerpo. Policleto (500 a.C) definió su canon de siete cabezas y media basándose en el sistema de proporciones matemáticas ideales, expresadas por los intervalos perfectos en la escala musical según Pitágoras. En el mismo canon definía la posición humana para hacer un reparto equilibrado del peso del cuerpo. La biomecánica actual hace, igualmente, estudios del reparto de peso en pisadas, pedales, y fuerzas diferentes en el estudio del deporte. También trata de alcanzar un ideal y cada entrenador, como cada artista, acomete la solución del problema de una manera.

Posteriores escultores, pintores y científicos han seguido usando o investi-

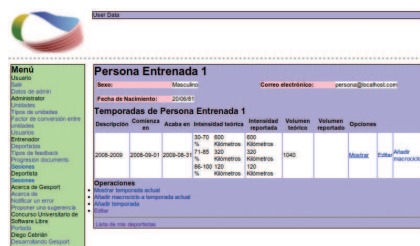


Figura 6: Pantalla del programa en funcionamiento. En concreto un resumen de la temporada con los datos acumulados.

gando estas proporciones, cambiándolas por la relación de 1 cuerpo a 8 cabezas, que aún se usa hoy. También se han incluido valores como la expresión del rostro (Lisipo, siglo IV a.C), que denotaban la importancia de la sensación interior del esculpido o estudiado. Gesport también proporciona campos suficientes como para hacer un análisis de lo que pasa por la cabeza del deportista.

Esta aplicación no hace otra cosa que darle al entrenador y escultor de mentes y cuerpos competidores, una herramienta para controlar que las proporciones de los diferentes tipos ejercicios sean las correctas, para ver los efectos del trabajo y para poder corregir las anomalías que surjan en su desarrollo. De esta manera se podrán alcanzar unos resultados, que, salvando las distancias con las esculturas clásicas, llenarán de belleza y emoción la vida, deportistas, entrenadores, y aficionados.

Author

Realizado por Diego Cebrián, estudiante de Ingeniería Técnica en Informática de Gestión por la Universidad de Valladolid, campus de Segovia. Ha trabajado siempre en sistemas con interfaz web diseñando informes, y sistemas de trazabilidad alimentaria, así como profesor de diseño gráfico, informática y realización de páginas web en proyectos de escuela taller, taller de empleo y cursos del INEM.

RECURSOS

- [1] Blog: gesport-blog.serincas.com
- [2] Programa en pruebas: gesport.serincas.com (Solicita tu cuenta al administrador)
- [3] Seguimiento del desarrollo: <https://forja.rediris.es/projects/cusl3-gesport/> (Solicita tu cuenta al administrador)
- [4] Documentalista histórico: David Díaz San Frutos

PIE

PIE (Plataforma de Información de Exámenes) es un proyecto para ayudar a los estudiantes y a los que se dedican a la enseñanza, principalmente centrado en el ámbito de la Universidad y Europa. La plataforma engloba material de varios tipos: técnicas de resolución de problemas, exámenes, apuntes y funciones de comunicación... en definitiva, a un nuevo marco de inteligencia.

La tradicional concepción de la inteligencia, muy presente en la educación superior dominante, requiere proponer nuevas alternativas. El concebirla como un enfoque simple y un proceso unitario se escapa de la realidad: un individuo; en los tiempos que corren, la inteligencia cambia cada día y la formamos todos.

El proyecto se encarga de mantener al día el factor humano más crítico gracias al uso de las redes sociales y su constante evolución, usa el software libre para que cualquiera pueda participar a cualquier nivel del mismo y mantiene una ventana abierta a una mejor evolución gracias a que está en contacto con todas las culturas en las que el estudio es importante.

Las tecnologías que usa son las mismas que en mis tres anteriores proyectos web, (mas las nuevas), y ayudan a la expansión del conocimiento sin obstáculos, ya sea vía móvil, vídeo, noticias, etc.

Autor

Juan Felipe Belón Pérez.

RECURSOS

- [1] Sitio web del proyecto: <http://pie.zenphp.es/>

GESTAS

GESTAS es un "GESTor de ASociaciones" libre orientado a cubrir las necesidades del 90% de las asociaciones: todas aquellas que no son demasiado pequeñas para no necesitar prácticamente gestión, ni suficientemente grandes para exigir un gran ERP (contabilidad analítica, gestión de materiales, ...)

La idea surgió en el seno del Grupo OPENTIA al comprobar que varias de las asociaciones en las que estamos involucrados necesitaban

una herramienta que les permitiera gestionar los datos de los socios, el pago de sus cuotas, la composición histórica de su junta directiva, sus votaciones y actas de reuniones, y, en general, toda su información oficial. Al no encontrar herramientas libres que cubrieran lo que buscábamos, decidimos crearla nosotros, y, dado que los dos desarrolladores principales aún seguimos ampliando nuestros estudios, decidimos presentarnos al Concurso Universitario de Software Libre por ser un marco estu-

Características Técnicas

GESTAS es una aplicación web colaborativa capaz de llevar la gestión de muchas asociaciones simultáneamente. Se puede instalar y administrar tanto en un servidor interno de la asociación como en modo servicio web, de manera que varias asociaciones puedan compartir sus costes de uso: servidor + alojamiento + conexiones + soporte y mantenimiento.

Su desarrollo se está realizando en un entorno LAMP (Linux + Apache + MySQL + PHP), aunque su diseño lo hace multiplataforma, funcionando tanto en otros sistemas operativos como con otros servidores web y gestores de base de datos. Además, con la idea de que la aplicación se pueda usar a nivel mundial, GESTAS se ha desarrollado para que sea multilingüe de base, haciendo uso de gettext y de la codificación de caracteres UTF-8.

El Futuro de GESTAS

En la actualidad GESTAS está en fase alfa y tiene lista una primera versión del núcleo y de la gestión de socios. Ahora se está ampliando la herramienta con el módulo de gestión de asociaciones. Le queda aún un gran camino por recorrer para que sea una herramienta completa y plenamente utilizable. Hay dos ramas principales por las que el proyecto irá desarrollándose, la de gestor de asociaciones en sí, en el que se desarrollaran módulos como el de



Figura 7: Captura de Pantalla de GESTAS. En esta pantalla se ve cómo el usuario administrador de una asociación acepta o no las peticiones pendientes a socios.

gestión de asociaciones, el gestor económico o el gestor de proyectos, y la de usabilidad, en la que se incluye desde un panel de control hasta la posibilidad de crear proyectos comunes entre asociaciones.

Por último, esperamos que, gracias a su diseño modular, se incorporen al proyecto otros módulos externos mediante *plugins*, pudiendo proporcionar funcionalidades que a nosotros ni siquiera se nos han pasado por la cabeza.

Autores

Pablo Álvarez de Sotomayor y Sergio Tomás Ortiz (Grupo OPENTIA).

RECURSOS

- [1] Servicio en productivo: <http://www.gestas.org>
- [2] Página del proyecto: <http://gestas.opentia.org>
- [3] Blog: <http://gestas.opentia.org/blog>
- [4] Forja: <https://forja.rediris.es/projects/cusl3-gestas/>
- [5] Listas de correo: https://forja.rediris.es/mail/?group_id=418
- [6] Patrocinador: <http://www.opentia.com>

AVBOT

Wikipedia es un proyecto que ha cosechado gran éxito gracias a su naturaleza colaborativa. Sin embargo, no todas las aportaciones son beneficiosas. AVBOT es un robot que corrige ediciones maliciosas.

¿Qué es Wikipedia?

Wikipedia es un proyecto para la creación de una enciclopedia libre que permite a los visitantes modifi-

car el contenido de las páginas, proporcionando un entorno de colaboración extrema. Sin embargo, es sensible a ataques, pues cualquier persona puede realizar modificaciones maliciosas. A estos cambios se les conoce como *vandalismos*.

El Problema del Vandalismo

A medida que Wikipedia ha ido incrementado su popularidad, se ha agravado el problema del vandalismo. Para solucionarlo se diseñaron herramientas semiautomáticas, pero seguía siendo una tarea tediosa para un humano. Como ya tenía experiencia en la programación de *bots* para Wikipedia, me propuse como reto realizar uno que revirtiese ediciones maliciosas de manera totalmente automática.

¿Cómo Funciona?

Cada vez que alguien modifica un artículo, aparece reflejado en una lista de *cambios recientes*, que podemos consultar. Con estos datos, el robot consulta el texto anterior y posterior al cambio, comparándolos. Entonces, un módulo de análisis comprueba la aparición de palabras malsonantes mediante una lista de expresiones regulares configurable. Cada expresión cuenta con un peso asociado, de modo que si la suma de los pesos provocados por una edición supera un determinado umbral, ésta se considera vandalismo y se revierte el artículo de Wikipedia a su estado anterior. Tanto las expresiones como los pesos han sido mejorados con la experiencia de varios meses de ejecución.

Otro tipo de edición maliciosa son los *blanqueos*, en los cuales un usuario elimina parcial o totalmente el contenido de un artículo. Esto es detectable calculando el porcentaje de texto retirado y revirtiendo si se supera cierta cifra.

Por cuestiones de eficiencia, dada la enorme cantidad de ediciones que recibe la Wikipedia en español (aproximadamente 30.000 al día), ha sido necesario realizar distintas optimizaciones al sistema, como el uso de hilos para paralelizar el trabajo y la precompilación de la lista de expresiones regulares.

Conclusiones

En estos últimos meses, ha acumulado más de 5000 horas de ejecución

y reparado más de 100.000 vandalismos, evitando una gran cantidad de horas de trabajo a la comunidad de Wikipedia en español. El proyecto ha resultado ganador en la Fase Local (Cádiz) del Tercer Concurso Universitario de Software Libre.

Autor

Emilio José Rodríguez Posada es Ingeniero Técnico en Informática de Sistemas y becario en la Oficina de Software Libre y Conocimiento Abierto de la UCA.

RECURSOS

[1] Más información sobre este proyecto:
<http://avbot.blogspot.com>

LongoMatch: The Digital Coach

LongoMatch es un proyecto de Software Libre diseñado para facilitar el análisis por vídeo de encuentros deportivos. Su función es la de ayudar a entrenadores y técnicos a realizar estudios técnico/tácticos de un partido, facilitando herramientas para las tres fases del estudio: marcado, análisis y presentación.

Motivación

En el deporte de alto nivel, el uso de las nuevas tecnologías es cada vez más habitual a la hora de intentar mejorar el rendimiento de un equipo o de un deportista. En los deportes de equipo con un alto componente táctico, como el hockey hierba, deporte al que juego desde pequeño, los entrenadores utilizan las grabaciones de partidos para analizar el juego del equipo contrario y poder plantear el encuentro de la mejor forma posible. Actualmente existen varios programas de pago con esta finalidad, pero su coste es desorbitado, en torno a los 6000 euros... un precio que mi club y otros muchos no se podían permitir. Por esa razón me decidí a desarrollar un proyecto de Software Libre para análisis de partidos, de tal forma que el dinero no fuese un impedimento para que todos pudiesen competir en igualdad de condiciones. La

idea del proyecto surgió hace ya un par de años y mi objetivo era crear una herramienta multiplataforma utilizando única y exclusivamente herramientas de Software Libre (Gtk+ para el entorno gráfico, db4o como base de datos y GStreamer como framework multimedia). Ha sido esta 3ª edición del Concurso Universitario de Software libre la que ha terminado de dar cuerpo al proyecto, materializando así la idea original.

Funcionamiento del Programa

Para hacer un estudio táctico de un partido es imprescindible poder localizar las jugadas más importantes para poder visualizarlas una y otra vez y analizar así el desarrollo del encuentro. LongoMatch permite marcar jugadas dentro de un archivo de vídeo y organizarlas por categorías (salidas, juego en el medio campo, defensa 1:1, defensa mixta, etc...), de tal forma que se puedan volver a visualizar con un simple click, pudiendo cambiar la velocidad de reproducción de la jugada para facilitar su estudio. En la Figura 1 se pueden ver los botones que se utilizan para marcar esas jugadas y la lista de jugadas que permite acceder a cada una de ellas. Además se puede ajustar "milimétricamente" el inicio y fin de cada jugada a través de un timeline, consiguiendo así delimitar la acción al segmento de interés. Una vez que se ha realizado el marcado de las jugadas importantes y el análisis del partido es necesario poder presentar los resultados, ya sea al cuerpo técnico o a los jugadores. Para ello LongoMatch permite crear listas de reproducción,

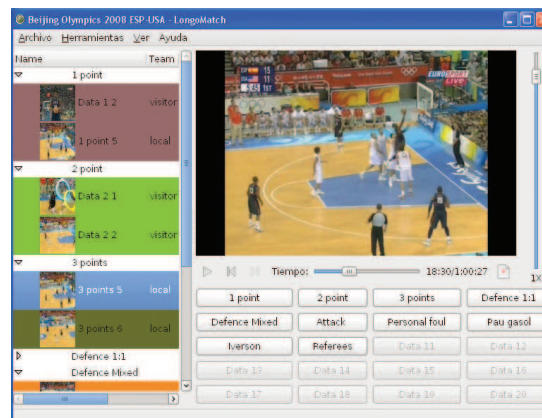


Figura 8: Ventana principal en la fase de marcado.

que a su vez se pueden exportar a un nuevo vídeo.

Presente y Futuro de LongoMatch

El proyecto ha tenido muy buena acogida, tanto en el mundo del deporte como en la comunidad del Software Libre. Cabe destacar que actualmente varios equipos de hockey, baloncesto o rugby, entre otros, están utilizando LongoMatch a nivel profesional. Además, LongoMatch ha sido uno de los primeros proyectos, junto con Elisa y Songbird, en apostar por el uso de GStreamer como framework multimedia en todas las plataformas. GStreamer es un gran veterano en sistemas *NIX, pero no proporciona binarios para Windows, dejando al desarrollador la difícil tarea de portabilidad y compilación. El esfuerzo por portar LongoMatch a Windows ha dado lugar a un nuevo proyecto (GStreamer-Winbuild) que mantengo junto a Andrés Colubri, siendo este el primero en ofrecer un instalador de GStreamer para Windows con soporte para un gran número de códecs, ofreciendo una alternativa libre a DirectShow.

La experiencia vivida en este concurso ha sido más que enriquecedora, habiendo recorrido todas las etapas en el desarrollo y mantenimiento de un proyecto de Software Libre. Animo a todos aquellos que tengan una idea a participar en el próximo concurso.

Autor

Andoni Morales Alastruey.

RECURSOS

[1] Web del proyecto: <http://longomatch.wordpress.com/>

OpenDomotica

OpenDomótica tiene como objetivo diseñar una red domótica libre, con un grupo de usuarios dispuestos a desarrollarla y hacerla más sólida cada vez. Las ideas básicas estaban: había que utilizar X10, ya que fue la primera tecnología domótica en aparecer (desde 1978), lo que hace que sea un estándar en el sector domótico. También tuvimos que decidir



Figura 9: Control de las luces de una casa desde Mister House.

qué soporte hardware utilizar, para ello elegimos Arduino, que además de ser libre, se ajusta perfectamente a las necesidades del proyecto.

Ya tenemos solucionado el protocolo y el soporte físico, ahora nos queda decidir nuestra aplicación para controlar el sistema domótico. Después de una evaluación nos decidimos a utilizar Mister House. Este es un popular paquete de software libre para el control domótico. Escrito en Perl, funciona bajo distintas plataformas, tales como Windows 95/98/NT/2K/XP y la mayoría de los sistemas basados en UNIX, incluyendo Linux y MacOS X. Con esto lo que conseguimos es un entorno domótico barato multiplataforma.

Facilidad del Sistema

Mister House nos brinda una interfaz web, por lo que no sólo podremos controlar nuestro sistema domótico de una manera fácil dentro de nuestra casa, sino que, estando en cualquier sitio del mundo, podremos controlar el estado de nuestra casa siempre que dispongamos de una conexión a internet. Todo el mundo puede imaginarse las ventajas de este sistema: seguridad, comodidad, ahorro de energía,...

En la Figura 1 podemos apreciar uno de los controles que nos ofrece Mister House, con ello os podéis hacer una idea de la potencia del mismo.

Motivación

Hoy en día la domótica está asociada al "lujo", y por tanto es sólo accesible para un pequeño sector de la población, cuando en realidad los costes de fabricación de un controlador domótico son muy inferiores a los de un teléfono móvil común. La idea es hacer el

máximo esfuerzo posible para que esto cambie, para que en un futuro no muy lejano todas las casas tengan un control domótico, por supuesto, libre.

¿Dónde se Encuentra el Proyecto?

De momento hemos conseguido controlar un Arduino utilizando Mister House, pero sólo a modo de prueba; está previsto que para finales de Marzo, principios de Abril esté terminada la librería de interconexión. Con esto ya podríamos montar el sistema utilizando Arduino.

Autor

Creado por Juan Antonio Infantes Díaz, estudiante de Ingeniería Técnica en Informática de Gestión en la Universidad de Málaga.

RECURSOS

[1] Para seguir el desarrollo del proyecto puedes visitar: <http://opendomotica.wordpress.com/>

Tucan Manager

Tucan es un proyecto de código abierto que pretende homogeneizar el acceso a los servicios de descarga directa, no sólo simplificando el uso a los usuarios que descargan, sino también a aquellos que suben contenidos.

Programado en Python y haciendo uso de las librerías gráficas GTK, es una aplicación de uso sencillo que se integra perfectamente en la mayoría de los escritorios Linux. Pero como no sólo de Linux vive el software libre, Tucan es multiplataforma y dispone también de un instalador para MS Windows.

Motivaciones

La idea tras los servicios como Megaupload o Rapidshare es bastante simple: "Quiero descargar, pero

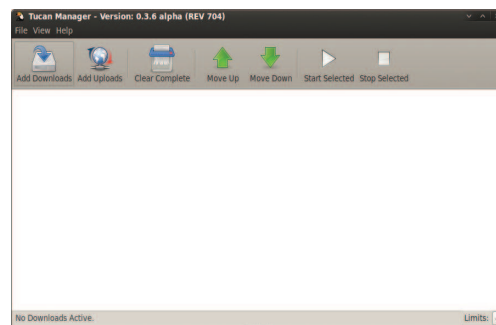


Figura 10: Tucan, listo para descarga.

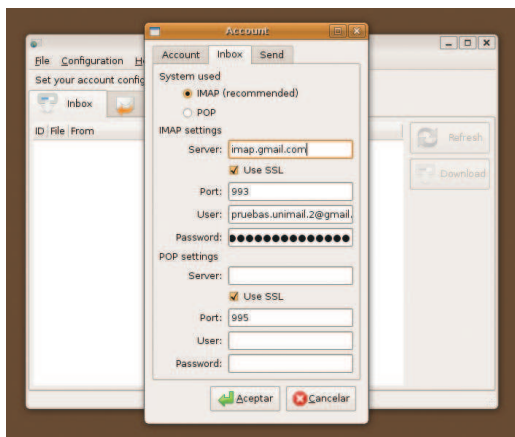


Figura 11: Ventana de configuración de Unimail.

quiero hacerlo ahora.” El usuario, normalmente, no desea comenzar una descarga hoy y que termine la semana próxima. Esto puede parecer una situación extrema, pero es algo común en redes p2p, más aún si lo que se pretende descargar es poco habitual.

El concepto de sistema distribuido subyacente en las redes p2p es muy interesante, pero al final las implementaciones suelen depender de servidores, ya sea para buscar nodos o para buscar contenidos.

Si tenemos que depender de servidores centralizados, ¿Por qué no aprovecharlos? La gran ventaja de los servicios de descarga directa no es que al descargar algo se tarde menos que usando p2p, ya que no siempre ocurre así, sino la de tener acotado el tiempo de descarga. No hay nada que desespere más a un usuario que una estimación engañosa sobre cuánto va a tardar algo, porque tendrá que estar pendiente de si ha terminado o no.

Estado del Proyecto

El desarrollo del proyecto Tucan es muy ambicioso, para el concurso se propuso implementar el gestor de descargas y subidas, pero están previstas más funcionalidades para el futuro.

- *Primera fase: Implementación de descargas.* Estas funcionalidades son, desde hace tiempo, estables y probablemente las que están haciendo de Tucan una aplicación popular en la comunidad del software libre.
- *Segunda fase: Implementación de subidas.* Actualmente se están implementando las subidas, algo

que probablemente no interese al usuario medio, pero que aumentará la cantidad de contenidos y el acceso a éstos.

- *Tercera fase: Implementación de búsquedas.* Para poder crear un motor de búsquedas, primero debemos finalizar las subidas.

Una vez tengamos una buena comunidad de “uploaders”, Tucan indexará los contenidos de las subidas y los pondrá a disposición de los usuarios. Imagina no tener que buscar

más enlaces en foros o blogs, sino tenerlo todo organizado y disponible en la misma aplicación que usarás para descargarlo. O poder suscribirte a una serie y que te lleguen avisos sobre nuevas publicaciones.

Ese es el objetivo de Tucan.

Autores

José Francisco Lupión González, Francisco Salido Ruiz y José Manuel Cordero Rodríguez.

RECURSOS

- [1] Pagina del proyecto: <http://cusl3-tucan.forja.rediris.es/>
- [2] Blog del proyecto: <http://tucaneando.wordpress.com/>
- [3] Documentación del proyecto: <http://cusl3-tucan.forja.rediris.es/doc/>

Unimail

Unimail es un programa de ordenador cuya finalidad es facilitar la transferencia de archivos por correo electrónico. Se trata de software libre y es multiplataforma, pues está hecho en Python y usa GTK+ para su interfaz de usuario gráfica, por lo que se puede ejecutar tanto en sistemas GNU/Linux, como en sistemas Windows, entre otros. Este proyecto participa en el III Concurso Universitario de Software Libre, que organiza la Universidad de Sevilla.

Algunas de sus opciones disponibles o planeadas a corto plazo son poder enviar un archivo en varias partes (saltándose el límite de tamaño de archivos adjuntos que imponen algunos servicios de correo electrónico), poder parar un envío o descarga en cualquier momento y

continuar el proceso más adelante, la comprobación automática de la integridad de un archivo recibido y poder encriptar el archivo a enviar de forma opcional.

Mi motivación principal para comenzar este proyecto fue el hecho de no disponer de una forma de enviar fácilmente un archivo de cualquier tamaño por correo electrónico. Por ejemplo, este problema lo he tenido siempre que he querido enviar las fotos de mi cámara en un archivo comprimido a algún amigo, ya que los servicios de correo electrónico suelen imponer límites de tamaño a la hora de adjuntar un archivo a un mensaje y, normalmente, las fotos que quiero enviar suelen tener, entre todas, un tamaño de decenas de megabytes (un tamaño que excede dichos límites). Esta motivación es sencilla, pero estoy concibiendo el proyecto como una herramienta que facilite varias tareas distintas a la hora de enviar un archivo, de forma que todo se haga en un solo paso.

El uso de Unimail por parte del usuario es muy sencillo. Una vez configurado el programa con los parámetros de la cuenta de correo electrónico (nombre de usuario, contraseña, etc.), el usuario sólo debe especificar el archivo que quiere enviar y el destinatario.

Es la primera vez que participo en el Concurso Universitario de Software Libre y, desde luego, no a va a ser la última. Esto se debe principalmente a que estoy aprendiendo mucho mientras desarrollo el proyecto, ya que lo comencé sin tener conocimientos sobre lo que estoy usando, como Python y GTK+.

Autor

José Antonio Jiménez Carmona es estudiante de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla. ■

RECURSOS

- [1] Blog de seguimiento del proyecto: <http://proyectounimail.wordpress.com>
- [2] Página web del proyecto: <https://forja.rediris.es/projects/cusl3-unimail>