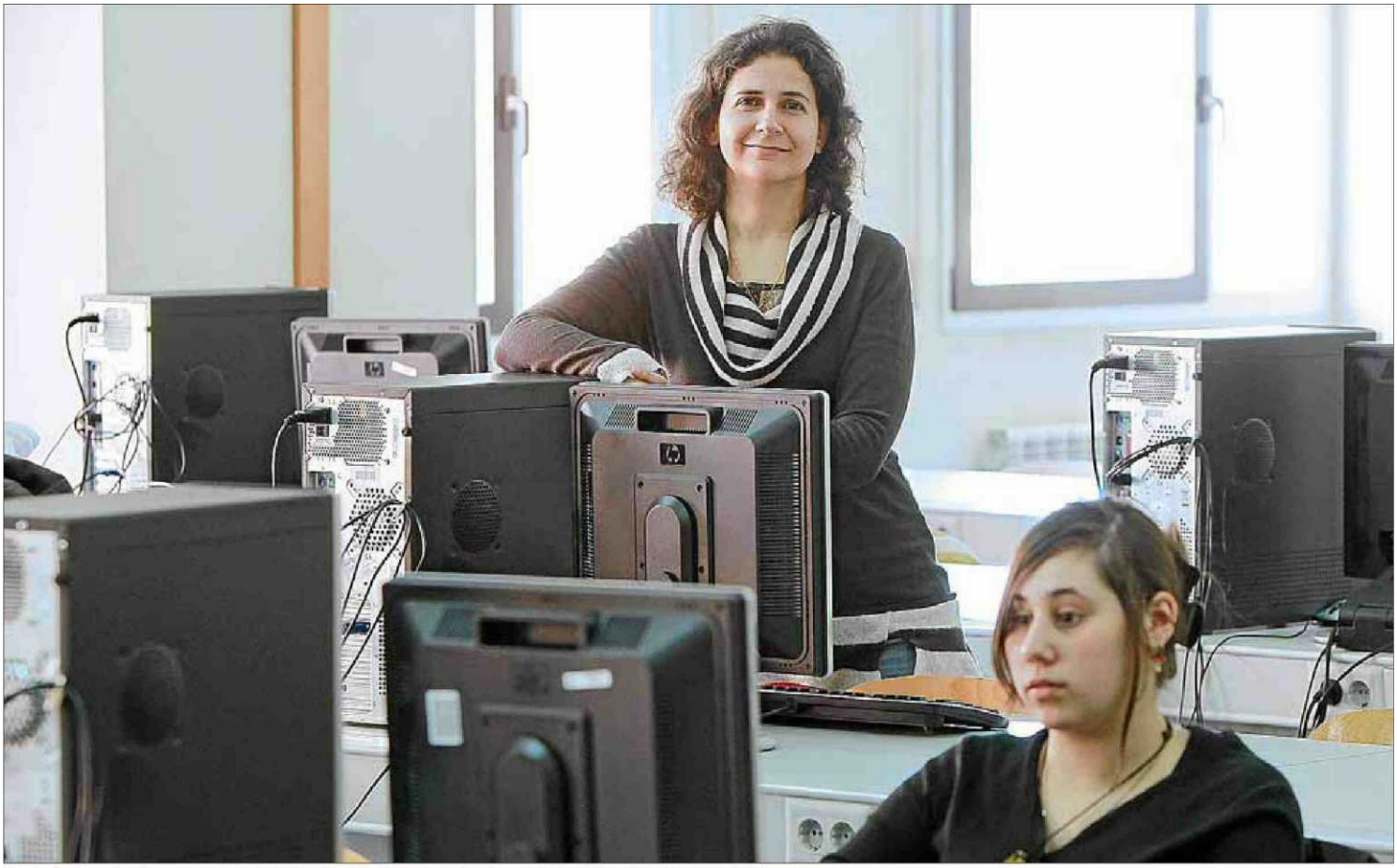
EL MUNDO DE CASTILLA Y LEON(INNOVADORES) 18/02/13
VALLADOLID

Prensa: Semanal (Lunes)
Tirada: 20.608 Ejemplares
Difusión: 15.581 Ejemplares

Cód: 67218148

Página: 7

Sección: OTROS Valor: 3.369,00 € Área (cm2): 768,0 Ocupación: 91,56 % Documento: 1/1 Autor: Núm. Lectores: 62324



Yania Crespo, de pie, en una sala de la facultad de Telecomunicaciones de la Universidad de Valladolid,. / J. M. LOSTAU

Los 'médicos' del software libre

El grupo de investigación Giro de la Universidad de Valladolid trabaja en la detección y solución de defectos informáticos, anticipándose a que suceda un problema para mejorar su calidad en el futuro. Por **Gemma Aliste**

Dice el proverbio que 'más vale prevenir que curar'. Y no solo en medicina, sino también en informática. Como si de un chequeo médico se tratara, se hace un análisis, se diagnostica y se actúa para prevenir 'enfermedades' en el *software* del futuro, anticipándose a los problemas.

El grupo de investigación GIRO de la Universidad de Valladolid lleva cuatro años trabajando en el progreso de ingeniería del *software*, dentro del apartado de evolución, donde se encargan de detectar defectos y aportar soluciones que serán beneficiosas a largo plazo. Hacen informática para la informática.

«Hay que arreglar algo antes de que pase», señala Yania Crespo, profesora contratada doctora de la Universidad de Valladolid (área de lenguajes y sistemas informáticos), que junto con otros cuatro profesores (Miguel Ángel Laguna y Esperanza Manso, de la UVA; Carlos López, de la Universidad de Burgos y Francisco Javier Pérez, de la Universidad de Amberes), lleva inmersa cuatro años en este proyecto. «El defecto que trabajamos es en un *software* libre que tiene problemas de cómo está hecho, lo cual hace posible que no evolucione bien», aclara. Esto es lo que ellos denominan detección, pero «no hay que quedarse ahí, hay que decir qué hacer» asegura.

Es ahí donde empieza la 'segunda' fase de su investigación: la refractorización. Consiste en hacer una 'operación' interna del *software* para que realice el mismo trabajo, pero diseñado por dentro de otra manera con el objetivo de eliminar el defecto y evitar así un problema de evolución en el tiempo. De no corregirse, realizar un cambio podría implicar una modificación completa del sistema afectado, aumentaría el tiempo de dedicación, e incluso, se correría el riesgo de «romper más cosas por el camino», destaca la profesora de la UVA.

Los investigadores aplican técnicas de inteligencia artificial y minería de datos

«Lo que más estamos trabajando ahora es cómo aprender de los desarrolladores, porque en detección de defectos hay una serie de reglas que van comprobando el código escrito y te avisan del problema que pueda surgir» explica, al igual que en una analítica donde el médico detecta alguna anomalía leve y recomienda prevención para evitar que se convierta en un problema mayor e intentar corregirlo sin consecuencias graves.

Es ese listado de alteraciones, a lo que en informática denominan falsos positivos o negativos. En este caso, puede ser que se avise de un defecto y no sea o que exista y no se comunique. «Reducir esos falsos positivos y negativos es la tarea fundamental que tenemos que desarrollar, porque lo que hemos visto en los últimos años es que haya o no, puede depender del contexto, de las herramientas con las que se está trabajando o incluso del tipo de software que se está haciendo, porque la misma regla no es igual para todos», asegura.

¿Cómo lo hacen? Aplicando técnicas de inteligencia artificial y minería de datos (clasificadores automáticos) para entrenar con retroalimentación de un desarrollador experto en ese ámbito y con los resultados obtenidos: «Este sí, este no, y con eso se vuelve a realizar el entrenamiento», explica. Estas técnicas de minería les permiten aprovechar el conocimiento de un desarrollador experto en este entorno para ajustar ese número de falsos positivos y negativos y poder reducirlos, para mejorar la calidad del *software* y ver cuáles son los mejores 'caminos'. De ahí, el nombre del proyecto, 'Road Map', una especie de mapa de carreteras en el cual ir siguiendo las 'rutas deseables'.

Construyen sus programas reutilizando elementos de otros que ya están inventados

La «clave fundamental» ha estado en ir construyendo software a base de bloques que se reutilizan. No reinventar, sino adaptar. «Siempre hemos trabajado en esta rama, hacemos mucho énfasis en la reutilización de cosas ya hechas, lo que hace que se aceleren los caminos. Ahora mismo podemos construir *software* de cinco millones de líneas de códigos porque gran parte está hecho ya», apunta Crespo.

Uno de los retos a los que se enfrentan ahora es el de trabajar con herramientas multilenguaje (diferentes 'alfabetos' dentro de un mismo proyecto), lo que ellos denominan «abordar la solución con cierta independencia del lenguaje», ya que cada uno tiene sus propias reglas y códigos.

A punto de poner fin a esta investigación, la titular de la UVA confiesa que han cumplido sus objetivos, pero «siempre son mejorables y necesitan ser corroborados en la práctica». «Continuaremos sin financiación hasta donde podamos, nuestra idea es aumentar la lista de defectos que somos capaces de mejorar», asegura.