

Beneficios de la aplicación del paradigma de líneas de productos *software* para generar *dashboards* en contextos educativos

(Benefits of the software product line paradigm in generating dashboards for educational contexts)

Andrea Vázquez-Ingelmo
Roberto Therón
Universidad de Salamanca, USAL (España)

DOI: <http://dx.doi.org/10.5944/ried.23.2.26389>

Cómo referenciar este artículo:

Vázquez-Ingelmo, A., y Therón, R. (2020). Beneficios de la aplicación del paradigma de líneas de productos *software* para generar *dashboards* en contextos educativos. *RIED. Revista Iberoamericana de Educación a Distancia*, 23(2), pp. 169-185. doi: <http://dx.doi.org/10.5944/ried.23.2.26389>

Resumen

Los datos son cruciales para mejorar la toma de decisiones y obtener mayores beneficios en cualquier tipo de actividad. Sin embargo, la gran cantidad de información generada debido a las nuevas tecnologías ha convertido el análisis de los datos y la generación de conocimiento a partir de ellos en una tarea compleja. Numerosas herramientas han surgido para facilitar esta generación de conocimiento, como es el caso de los *dashboards* o paneles de información. Aunque los paneles de control sean herramientas muy potentes, su efectividad puede verse afectada por un mal diseño o por no tener en cuenta el contexto en el que se encuadran. Por ello, es necesario diseñar y crear paneles de control a medida en función de la audiencia y dominio de los datos. Crear paneles de control personalizados puede ser muy beneficioso, pero también un proceso costoso en lo que al tiempo y recursos se refiere. Este trabajo presenta una aplicación del paradigma de líneas de productos *software* para generar paneles de control adaptados a cualquier contexto de manera más sencilla, reutilizando tanto componentes *software* como conocimiento. Uno de los contextos que puede verse especialmente favorecido por este enfoque es el contexto educativo, donde la analítica del aprendizaje y el análisis de datos sobre el rendimiento de los estudiantes se está popularizando. Contar con paneles de control personalizables para cualquier rol (estudiante, profesor, administrador, etc.) puede mejorar los procesos de toma de decisiones, mostrando a cada usuario la información que más le interesa de la forma que mejor le permita comprenderla.

Palabras clave: tecnología educativa; investigación educativa; alfabetización en datos.

Abstract

Data are crucial to improve decision-making and to obtain greater benefits in any type of activity. However, the large amount of information generated by new technologies has made data analysis and knowledge generation a complex task. Numerous tools have emerged to facilitate this knowledge generation, such as dashboards. Although dashboards are very powerful tools, their effectiveness can be affected by a bad design or by not taking into account the context in which they are placed. Therefore, it is necessary to design and create tailored dashboards according to the audience and data domain. Creating tailored dashboards can be very beneficial, but also a costly process in terms of time and resources. This paper presents an application of the software product line paradigm to generate dashboards adapted to any context in a more straightforward way by reusing both software components and knowledge. One of the contexts that can be especially favored by this approach is the educational context, where Learning Analytics and the analysis of student performance to improve learning methodologies are becoming very popular. Having tailored dashboards for any role (student, teacher, administrator, etc.) can improve decision making processes by showing each user the information that interests them most in the way that best enables them to understand it.

Keywords: educational technology; educational research; data literacy

Los datos han comenzado a ser cruciales en cualquier actividad, ya sean actividades profesionales o cotidianas. Realizar el proceso de toma de decisiones con una base robusta de información es crucial para potenciar los resultados positivos de esta tarea (Albright, Winston y Zappe, 2010). Los frutos de esta toma de decisiones son una serie de acciones, acciones que buscan reportar algún tipo de beneficio en el contexto de aplicación.

Sin embargo, llevar a cabo una toma de decisiones dirigida por datos (Patil y Mason, 2015) no es una tarea trivial. Primero, porque es necesaria una cantidad significativa de datos para poder generar conocimiento. Y segundo, porque los procesos de análisis de dichos datos requieren que la persona al frente de la toma de decisiones o del análisis sea capaz de comprender e interpretar conjuntos de datos que en muchas ocasiones son complejos y extensos.

No obstante, la evolución de las tecnologías ha permitido que estas tareas de análisis estén al alcance de perfiles menos técnicos. Existen herramientas que facilitan el análisis y la generación de conocimiento a partir de conjuntos de datos. Una de las herramientas que más se han popularizado son los *dashboards* o paneles de control (Sarıkaya, Correll, Bartram, Tory y Fisher, 2018).

Los paneles de control son elementos digitales compuestos por una serie de visualizaciones de datos. Estas visualizaciones tratan de codificar información a

través de las propiedades de sus elementos visuales, como la posición, los colores, los tamaños, las formas, etc. (Munzner, 2014).

Pero contar con un panel de control no garantiza la generación de conocimiento. Es necesario tener en cuenta la audiencia y el perfil de los usuarios que utilizarán estas herramientas. Pueden existir usuarios que puedan comprender visualizaciones complejas, mientras que otros necesitarán otras metáforas visuales para entender correctamente sus conjuntos de datos (Aldrich y Sheppard, 2000).

Así pues, los paneles de control son herramientas potentes, pero necesitan un elaborado proceso de diseño y desarrollo para cumplir con las expectativas de sus usuarios y para ser eficaces a la hora de generar conocimiento. Además, en contextos en los que existen diversos roles y perfiles de usuario, este proceso de diseño es aún más complejo, dado que hay que tener en cuenta muchos más factores. Una solución sería realizar un panel de control hecho a medida para cada perfil implicado, pero sería prácticamente imposible de llevar a cabo debido a la cantidad de tiempo y recursos que serían necesarios.

En la literatura se han considerado diversos enfoques para permitir la automatización de este proceso de diseño e implementación de paneles de control, con el objetivo de disminuir el tiempo necesario para desarrollar paneles de control hechos a medida (Vázquez-Ingelmo, García-Peñalvo y Therón, 2019c).

Entre estos enfoques se encuentran desde asistentes gráficos de configuración que permiten a los usuarios elegir qué gráficos formarán parte de sus paneles de control, hasta desarrollo dirigido por modelos que consiguen generar paneles de control personalizados basándose en descripciones formales del dominio (Kintz, Kochanowski y Koetter, 2017; Logre, Mosser, Collet y Riveill, 2014; Palpanas, Chowdhary, Mihaila y Pinel, 2007; Pleuss, Wollny y Botterweck, 2013), entre otras variadas soluciones.

Entre estas soluciones se encuentra el paradigma de las líneas de productos *software* (Clements y Northrop, 2002; Gomaa, 2004). Este paradigma analiza el dominio de los productos que quieren desarrollarse para encontrar similitudes y diferencias entre ellos, de tal modo que puedan construirse productos a través de la composición de diversas características. Este enfoque se ajusta perfectamente al dominio de los paneles de control, dado que son herramientas que comparten varias propiedades.

En este trabajo se presentan y discuten los beneficios de la aplicación de este paradigma al dominio de los paneles de control, especialmente en contextos educativos, en los que diversos perfiles están involucrados. El objetivo principal es acercar la visualización de datos a cualquier usuario, así como fomentar la generación de conocimiento. En un contexto educativo, contar con paneles de control hechos a medida para cada actor involucrado podría significar una explotación más eficaz y efectiva de los datos de aprendizaje generados y obtenidos a través de diversas fuentes.

LOS PANELES DE CONTROL EN EL CONTEXTO EDUCATIVO

Como se ha mencionado en la introducción, los paneles de control son herramientas cada vez más populares, debido a su utilidad a la hora de soportar el análisis visual de conjuntos de datos complejos. El contexto educativo es uno de los contextos en los que estas herramientas pueden traer beneficios significativos, dado que el uso de datos para tomar decisiones relativas a los procesos educativos puede mejorar el aprendizaje (Cooper, 2007; Mandinach y Honey, 2008).

Los paneles de control educacionales (Yoo, Lee, Jo y Park, 2015) son instrumentos que permiten a sus usuarios identificar patrones, relaciones, datos relevantes, etc., entre un conjunto de variables de aprendizaje.

Sin embargo, en un contexto como el educativo, muchos roles pueden verse involucrados: desde los propios estudiantes, hasta profesores, jefes de estudio o directores, entre otros. Como es evidente, estos roles tendrán diversos objetivos a la hora de explorar sus datos, dependiendo de sus necesidades.

Esta diversidad de roles se analizó en una revisión de la literatura realizada por Schwendimann et al. (2017) en cuanto a los paneles de control educativos. En dicha revisión de la literatura se observó que la mayoría de los usuarios suelen ser profesores, pero también se encuentran estudiantes, administradores e investigadores entre los principales usuarios de estas herramientas (Schwendimann et al., 2017). Los paneles de control educativos también son diversos en lo que respecta a sus objetivos; monitorización propia, monitorización de otros estudiantes y monitorización administrativa (Schwendimann et al., 2017).

La mencionada revisión de la literatura también muestra los principales tipos de gráficos o visualizaciones utilizadas para mostrar información sobre el aprendizaje en función del rol del usuario. Así pues, los gráficos más utilizados en general por todos los roles son los diagramas de barras, los diagramas de líneas y las tablas, lo que puede deberse a su simplicidad.

Este tipo de investigaciones permiten observar que los paneles de control son muy diversos en el contexto educativo, tanto en sus funcionalidades como en su diseño, dado que estas características son las que definen el fin del instrumento (y su eficiencia).

Debido a estos factores, se han buscado métodos y realizado propuestas para diseñar paneles de control educativos y de analítica del aprendizaje para que puedan ser adaptados según sus propósitos y audiencia.

No existe un enfoque único que sirva para todos los usuarios (Teasley, 2017). En el contexto educativo, los paneles de control no solo buscan informar a tutores sobre el rendimiento de los alumnos, sino que también pueden convertirse en herramientas que motiven a estos últimos. Pueden incluso servir como instrumentos para que los estudiantes se auto-regulen y puedan comparar sus propios resultados. Sin embargo, puede que no todos los estudiantes respondan de la misma forma ante la información mostrada en un panel de control sobre su rendimiento (Teasley, 2017).

Los paneles de control deben personalizarse para ofrecer la información necesaria de la manera más efectiva. De hecho, en un estudio realizado por Roberts, Howell y Seaman (2017) se confirmó el deseo generalizado de los estudiantes por contar con paneles de control que puedan ser personalizados a su gusto, dándoles opción de configurarlos para mostrar la información que más les interesa o que ven más útil.

Así pues, no es solo la variedad de roles de usuarios en el contexto educativo, sino la variedad de objetivos y perfiles entre usuarios con un mismo rol, lo que convierte el desarrollo de paneles de control que presentan analíticas del aprendizaje en una actividad elaborada. Sumado a todo ello, la cantidad de datos generados y su compleja estructura pueden dificultar aún más el proceso de descubrimiento de conocimiento por parte de perfiles menos técnicos.

Por estas razones, se han propuesto modelos para intentar adaptar estas herramientas usando modelos conceptuales en los que se tienen en cuenta los indicadores, la descripción y necesidades de los usuarios, sus preferencias, su conocimiento del dominio, etc. De hecho, en Dabbebi, Iksal, Gilliot, May y Garlatti (2017) se presenta un generador de paneles de control de analítica del aprendizaje que tiene en cuenta la mencionada información. Esta información es estructurada en modelos que alimentan un generador que permite obtener paneles de control adaptados al perfil de usuario y tareas descritas en los modelos.

Como puede observarse, los paneles de control en el contexto educativo han incrementado su popularidad debido a los beneficios que puede suponer el uso de los datos en la toma de decisiones (Patil y Mason, 2015). Sin embargo, para sacarles partido, es necesario tener en cuenta a los usuarios y el contexto en el que serán empleados, y realizar pruebas de aceptación para comprobar que dichas herramientas pueden mejorar los procesos de aprendizaje.

METODOLOGÍA

Como se ha introducido, el desarrollo de paneles de control hechos a medida supone un reto, tanto a nivel de diseño como de implementación. Por ello, es necesario contar con paradigmas que potencien la productividad a la hora de desarrollar estas herramientas.

El paradigma de líneas de productos *software* permite identificar, a través de ingeniería de dominio y la abstracción de este, puntos comunes entre los sistemas que conforman el espacio de posibles productos. Esta identificación de similitudes no solo sirve para reutilizar código y disminuir los tiempos de implementación, sino también para identificar y estructurar factores importantes que afectan al diseño de los paneles de control.

Las líneas de producto *software*

Las líneas de productos *software* es uno de los paradigmas de reutilización sistemática de código más extendidos y aplicados en la práctica y en contextos reales. Es especialmente su viabilidad práctica lo que coloca esta metodología como una potente herramienta para abordar el desarrollo de software de manera masiva.

Este paradigma se compone de dos fases. La primera, denominada fase de ingeniería de dominio (Pohl, Böckle y Van der Linden, 2005), es clave para la creación de líneas de productos *software*. Durante esta fase se identifican las principales propiedades y características que tendrán los productos pertenecientes a la familia. Estas características representan las similitudes y diferencias entre los diversos productos del dominio. La identificación de propiedades abstractas del dominio permite formalizar los componentes o *core assets* necesarios para su implementación.

En esta fase también se definen los denominados puntos de variabilidad. Los puntos de variabilidad son la forma de introducir, modificar o adaptar ciertas funcionalidades según los requisitos del producto a desarrollar (Pohl, Böckle y Van der Linden, 2005). Esta tarea es esencial, ya que es la que permite materializar las características del dominio en componentes *software*. Existen diversos métodos para modelar los mencionados puntos de variabilidad (Metzger y Pohl, 2007; Van Gurp, Bosch y Svahnberg, 2001).

Uno de los mecanismos más extendidos para representar la variabilidad en las líneas de productos software es el método FODA (*Feature-Oriented Domain Analysis*) (Kang, Cohen, Hess, Novak y Peterson, 1990). Los diagramas de características (*feature models*) se introducen por primera vez en (Kang et al., 1990) como mecanismo formal de descripción de las propiedades del dominio. A través de estos diagramas jerárquicos pueden especificarse las características obligatorias, alternativas, opcionales, etc., de la familia de productos, o lo que es lo mismo, sus puntos de variabilidad.

Una vez estudiado el dominio e identificadas las características de la línea, la siguiente fase es la fase de ingeniería de aplicación. Haciendo uso de la información y recursos obtenidos en la fase anterior, se crean instancias concretas de la línea de productos, seleccionando las características que compondrán el producto específico a construir (Gomaa, 2004).

En función de la configuración seleccionada a través de los modelos instanciados, se configurarían y compondrían los componentes base (*core assets*) para implementar el producto final. Gracias a su implementación durante la fase de ingeniería del dominio, los componentes base estarían preparados para ser configurables, con lo cual solo sería necesario adaptarlos a los requisitos específicos, demostrando la potencia de este paradigma.

Generación de código

Existen diversas formas de inyectar variabilidad en los componentes base de la familia de productos. La generación o instanciación automática de productos de la familia podría verse como una combinación del paradigma de las líneas de productos y el desarrollo dirigido por modelos (Anquetil et al., 2008).

La división de los productos en una serie de componentes configurables con características definidas posibilita la creación de generadores de código que seleccionen y configuren acordeamente los componentes base para obtener el código fuente del producto final.

Para conseguir esto, es necesario materializar los puntos de variabilidad obtenidos en la fase de ingeniería del dominio en el propio código fuente de los componentes base. El rol del generador de código consistiría en inyectar los parámetros de configuración específicos para adaptarlos a los requisitos del producto a generar.

Existen diversas técnicas para materializar estos puntos de variabilidad en el código fuente, por ejemplo: delegación de funcionalidades, herencia, parametrización, sobrecarga de métodos o funciones, bibliotecas de enlace dinámico (DLL) o compilación condicional (Gacek y Anastasopoulos, 2001).

Las plantillas de código son otra técnica muy extendida para abordar este problema (Tajali, Corriveau y Shi, 2013). Éstas permiten definir partes estáticas y partes dinámicas en las que puede inyectarse código en función de una serie de reglas (Greifenberg et al., 2016; Magdaleníć, Radošević y Kermek, 2011).

Para esta investigación se han utilizado plantillas como método de generación de código, debido a que es un método viable para las líneas de productos *software* (Greifenberg et al., 2016; Tajal, Corriveau y Shi, 2013), un método ampliamente utilizado en el desarrollo web (Kirda y Kerer, 2000; Tsubori y Suzumura, 2009) y permiten la introducción de sentencias condicionales y bucles, haciendo sencilla la materialización de los puntos de variabilidad a partir de modelos. Además, las plantillas permiten lograr un grado de granularidad muy fino en lo que se refiere a la materialización de características, lo que es altamente beneficioso en un dominio tan complejo como el de los paneles de control (Vázquez-Ingelmo, García-Peñalvo y Therón, 2019a).

En cuanto a la generación de código, se ha utilizado la siguiente estructura: el generador de código toma la configuración del panel de control previamente especificada a través de ficheros de configuración XML (cuya sintaxis está basada en las características abstractas del dominio previamente identificadas), inyectando la información de configuración en las plantillas de código que, en este caso, conformarán los componentes base de la línea de productos de paneles de control.

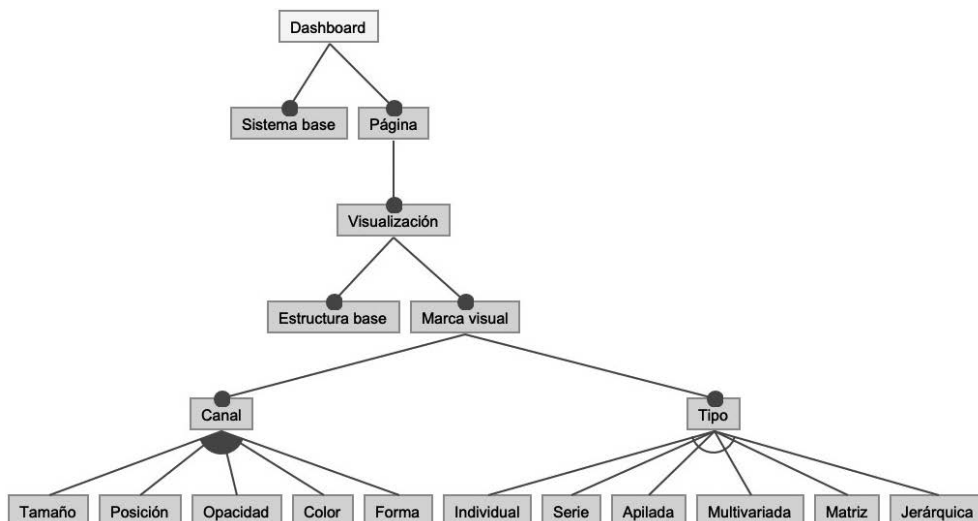
Las plantillas han sido realizadas con el lenguaje de plantillas Jinja2 (Ronacher, 2008). Este método es independiente al marco de desarrollo, y, aunque deba ser utilizado junto al lenguaje de programación Python, es posible inyectar información externa y generar cualquier tipo de archivos (Clark, 2018).

Cada componente gráfico de los paneles de control identificado mediante ingeniería de dominio ha sido implementado de manera individual, de tal forma que estos elementos primitivos puedan ser compuestos a través de plantillas Jinja2 para obtener visualizaciones concretas.

RESULTADOS PRELIMINARES

Gracias a la aplicación del enfoque de las líneas de productos *software* al dominio de los paneles de control, ha sido posible obtener una serie de componentes base (*core assets*) que pueden ser combinados para obtener paneles de control funcionales.

Figura 1. Diagrama de características simplificado de la línea de productos *software* de paneles de control



Como puede observarse en la figura 1, un panel de control estará compuesto por un sistema base y una o más páginas. Estas páginas contendrán una serie de visualizaciones, que contarán con una estructura base común y un conjunto de marcas visuales. Las marcas visuales pueden ser de distintos tipos según los datos que quieran representarse (marcas individuales, serie, apiladas, multivariadas, jerárquicas o matrices). Para codificar la información, se utilizarán uno o más canales, que se corresponden con propiedades de la marca visual, como su color, posición, tamaño, opacidad, forma, etc.

La figura 1 muestra una versión simplificada del diagrama con el objetivo de facilitar la comprensión del mismo, pero las visualizaciones contienen más elementos básicos, como sus ejes, leyendas, patrones de interacción, etc. (Vázquez-Ingelmo, García-Holgado, García-Peñalvo y Therón, 2019; Vázquez-Ingelmo, García-Peñalvo y Therón, 2019b; Vázquez-Ingelmo, García-Peñalvo, Therón y Conde González, 2019).

Estas pequeñas piezas abstractas mostradas en el diagrama de características tienen un grano fino de configuración, lo que permite personalizar el panel de control no solo a nivel global, sino también a nivel de visualización.

Para ilustrar la línea de productos obtenida mediante un ejemplo, se ha utilizado parte de los registros del conjunto de datos recogido en (Kuzilek, Hlosta y Zdrahal, 2017), el cual contiene datos demográficos, de las interacciones y del rendimiento de los estudiantes de Open University (OU).

Por ejemplo, a través de un fichero de configuración como el mostrado en la figura 2, es posible definir un panel de control concreto basado en las propiedades abstractas del dominio. Este fichero utiliza la tecnología XML para estructurar las características de los paneles de control a generar. Por ejemplo, el panel de control descrito en la figura 2 tendrá como título “*Learning Dashboard*”, y contará con una página en la que se encuentra un componente de tipo visualización donde se mostrarán las puntuaciones de tareas por estudiante a través de marcas visuales de tipo “barra”. Toda esta sintaxis está basada en el diagrama de características mostrado en la figura 1.

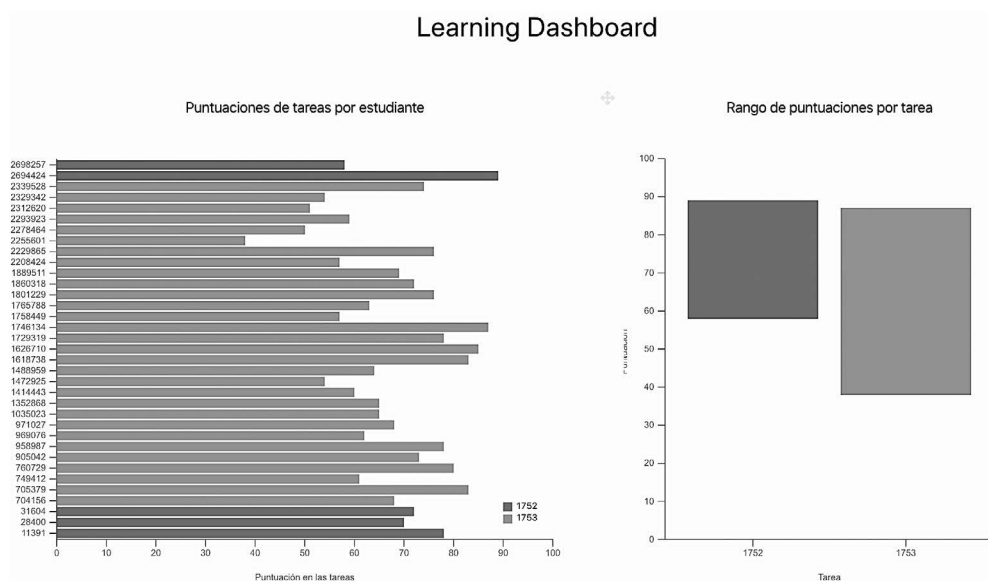
Figura 2. Fragmento de un fichero de configuración utilizado para generar un panel de control

```
<Dashboard>
  <Title>Learning Dashboard</Title>
  <Dataset>
    <Path>studentAssessment.csv</Path>
    <Type>CSV</Type>
  </Dataset>
  <Page page_id="1">
    <Component type="visualization" component_id="1">
      <Position>
        <x>0</x>
        <y>0</y>
        <width>5</width>
        <height>8</height>
      </Position>
      <Title>Puntuaciones de tareas por estudiante</Title>
      <Primitives>
        <Axis type="y" linear="true" multi="false">
          <Scale>
            <Accessor>id_student</Accessor>
          </Scale>
        </Axis>
        <Axis type="x" linear="true" multi="false">
          <Scale>
            <Accessor>score</Accessor>
            <Domain>percentage</Domain>
          </Scale>
          <Label>
            Puntuación en las tareas
          </Label>
        </Axis>
        <Legend>
          <Scale>
            <Accessor>id_assessment</Accessor>
            <ScaleType>band</ScaleType>
          </Scale>
          <Position>bottom-right</Position>
        </Legend>
        <Mark type="individual" linear="true">
          <Shape>bar</Shape>
          <Channels>
            <Position_Y>
              <Accessor>id_student</Accessor>
            </Position_Y>
            <Position_X>
              <Accessor>score</Accessor>
              <Domain>percentage</Domain>
            </Position_X>
            <Color>
              <Accessor>id_assessment</Accessor>
            </Color>
          </Channels>
        </Mark>
      </Primitives>
    </Component>
  </Page>
</Dashboard>
```

Una vez definidas las características del panel de control y de sus componentes (visualizaciones, canales de codificación, estilos), el fichero de configuración es utilizado por el generador de código para instanciar el panel de control a través de la inyección en las plantillas de código de los parámetros especificados. El resultado son los ficheros fuente del panel de control con las características establecidas (figura 3).

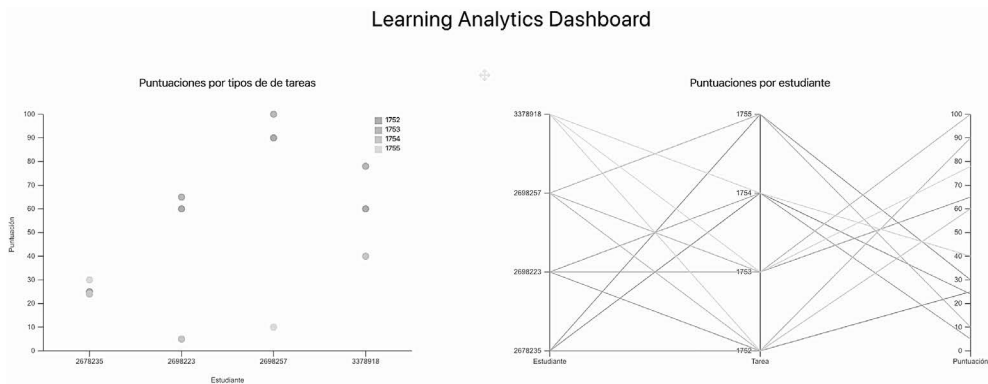
En este caso, el panel de control contiene dos visualizaciones que representan las puntuaciones de las tareas realizadas por cada estudiante y el rango de puntuaciones (mínimo y máximo) por tarea, respectivamente.

Figura 3. Ejemplo de un panel de control generado a través de su definición XML



Cambiar el tipo de gráfico o metáforas visuales para mostrar los datos en una visualización es tan sencillo como modificar sus primitivas en el fichero de configuración, lo que permite probar nuevas estructuras o composiciones para el panel de control sin preocuparse de su implementación. Por ejemplo, la figura 4 muestra otra composición para el panel de control utilizando un diagrama de dispersión y unas coordenadas paralelas para codificar la información relativa a las puntuaciones de las tareas de los estudiantes.

Figura 4. Ejemplo de un segundo panel de control generado a través de otra definición XML



DISCUSIÓN

La creación de paneles de control es una tarea compleja, sea cual sea el contexto en el que estén enmarcados, dado que la generación de conocimiento es crucial para mejorar los procesos de negocio. En lo que se refiere al ámbito educativo, este tipo de herramientas se vuelven especialmente útiles para generar conocimiento respecto al rendimiento de los estudiantes o la efectividad de los métodos de enseñanza utilizados.

Utilizar métodos para disminuir los tiempos de desarrollo de los paneles de control no solo reporta beneficios a la hora de crear estos instrumentos, sino que, además, permite generar paneles de control personalizados con menor esfuerzo, pudiéndose dedicar mayor tiempo y recursos a la fase de elicitación de requisitos.

La elicitación de requisitos es una fase esencial en cualquier proceso de desarrollo *software*, pero en el caso de los paneles de control toma especial relevancia, debido a que una mala interpretación de estos puede comprometer la efectividad de estas herramientas. Contar con un robusto sistema de generación de paneles de control facilita la realización de pruebas con usuarios (dado que pueden crearse prototipos rápidamente) y permite ejecutar cambios en la configuración de estos sin consumir muchos recursos.

Como se ha mencionado anteriormente, el contexto educativo puede ser un claro beneficiario de la aplicación de las líneas de productos *software* al dominio de los paneles de control. La cantidad de datos de aprendizaje que son generados debido a la popularización de nuevas tecnologías en la educación (Ferguson, 2012) hace necesario contar con nuevos métodos e instrumentos que permitan obtener beneficios de dicha información.

Si bien los paneles de control son herramientas muy útiles para estos procesos de análisis, es necesario tener en cuenta la audiencia que los utilizará, sobre todo

en ámbitos educativos donde los roles y los perfiles de usuario pueden ser muy heterogéneos en cuanto a objetivos, características y preferencias (Schwendimann et al., 2017).

La analítica del aprendizaje busca la mejora de los procesos de aprendizaje desde diversos puntos de vista (Long y Siemens, 2011); mientras que los educadores pueden beneficiarse de los datos para mejorar sus planificaciones y metodologías, los estudiantes pueden ver su progreso e incluso motivarse a través de los datos recolectados. Por otro lado, a más alto nivel, los administradores pueden utilizar la información para asignar recursos o planificar presupuestos de una forma más informada (Long y Siemens, 2011).

Poder generar paneles de control de manera sencilla, dedicando más tiempo al diseño y conceptualización del panel de control que a su implementación, permite tener productos mejor diseñados y adaptados a situaciones concretas en menor tiempo, así como responder de manera rápida y adecuada a la continua evolución que presentan los datos de aprendizaje (Liñán y Pérez, 2015; Picciano, 2012).

A su vez, contar con un generador de código permite realizar un análisis previo de los datos que permite inferir ciertas características de los paneles de control antes de ser generados, asegurándose de una correcta elección de las propiedades visuales antes de obtener el código fuente (por ejemplo, una variable de tipo nominal no puede codificarse usando una escala cuantitativa). De esta forma, es posible descargar a los usuarios de tareas más técnicas, como elegir las escalas o dominios a utilizar para representar una determinada variable.

Sin embargo, este enfoque sigue requiriendo de un riguroso análisis de los requisitos y perfiles de cada usuario, puesto que, de lo contrario, el panel de control generado puede resultar poco útil o efectivo para sus propósitos.

Uno de los siguientes pasos para conseguir un sistema más potente es implementar la capacidad de inferir automáticamente las características de los paneles de control a partir del perfil del usuario que lo empleará. De este modo, podrían presentarse gráficos o metáforas visuales más simples a un profesor con baja alfabetización en visualización de datos, mientras que un usuario más avanzado podría contar con visualizaciones más complejas en su pantalla, siempre de acuerdo con las tareas y objetivos que éstos tengan respecto a sus datos.

La toma de decisiones dirigida por datos puede llegar a ser muy útil, pero es necesario tener siempre en cuenta el contexto de aplicación y comprobar de manera continua que la implantación de estas herramientas y metodologías realmente funciona y trae beneficios a las personas involucradas.

CONCLUSIONES

El paradigma de las líneas de productos *software* ha sido aplicado al dominio de los paneles de control de información. El principal propósito de esta aplicación es disminuir la complejidad del proceso de diseño y creación de estas herramientas.

Mediante la abstracción del dominio, ha sido posible obtener una serie de componentes abstractos parametrizables que pueden ser combinados para generar paneles de control específicos con menor consumo de recursos.

Este enfoque puede aplicarse a cualquier contexto, en especial el educativo, donde el análisis de datos del aprendizaje se está popularizando de cara a mejorar los procesos y métodos de aprendizaje. Ofrecer paneles de control personalizados a los actores involucrados en estos procesos podría facilitar y acercar el análisis de datos a cualquier perfil de usuario.

Las líneas de investigación futuras incluyen la realización de pruebas con usuarios para obtener información sobre qué tipos de visualizaciones son más efectivas en función del contexto de los datos y las características de los usuarios, para incorporar estos datos al generador de código e inferir configuraciones efectivas automáticamente.

AGRADECIMIENTOS

Esta investigación ha sido parcialmente financiada por el Ministro de Economía y Competitividad del Gobierno de España a través del Proyecto DEFINES (TIN2016-80172-R). Este trabajo ha sido apoyado por el Ministerio de Educación, Cultura y Deporte bajo una beca FPU (FPU17/03276).

REFERENCIAS

- Albright, S. C., Winston, W., y Zappe, C. (2010). *Data analysis and decision making*. Cengage Learning.
- Aldrich, F., y Sheppard, L. (2000). Graphicacy-the fourth'R'? *Primary Science Review*, 64, 8-11.
- Anquetil, N., Grammel, B., Galvão, I., Noppen, J., Khan, S. S., Arboleda, H., . . . Garcia, A. (2008). Traceability for model driven, software product line engineering. Ponencia presentada en ECMDA Traceability Workshop Proceedings.
- Clark, S. (2018). Render your first network configuration template using Python and Jinja2. *Cisco Blogs*. Recuperado de <https://blogs.cisco.com/developer/network-configuration-template>
- Clements, P., y Northrop, L. (2002). *Software product lines*. Addison-Wesley.
- Cooper, M. M. (2007). Data-driven education research. *Science*, 317(5842), 1168-1171. doi: 10.1126/science.317.5842.1171.
- Dabbebi, I., Iksal, S., Gilliot, J.-M., May, M., y Garlatti, S. (2017). Towards Adaptive Dashboards for Learning Analytic: An Approach for Conceptual Design and implementation. Ponencia presentada en 9th International Conference on Computer Supported Education (CSEDU 2017), Porto, Portugal, (pp. 120-131). doi: 10.5220/0006325601200131.
- Ferguson, R. (2012). Learning analytics: Drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5/6), 304-317. doi: 10.1504/IJTEL.2012.051816.
- Gacek, C., y Anastasopoulos, M. (2001). Implementing product line variabilities. Ponencia presentada en ACM SIGSOFT Software Engineering Notes, New

- York, NY, USA. 26, (pp. 109-117). doi: 10.1145/375212.375269.
- Gomaa, H. (2004). *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley Longman Publishing Co., Inc.
- Greifenberg, T., Müller, K., Roth, A., Rumpe, B., Schulze, C., y Wortmann, A. (2016). Modeling variability in template-based code generators for product line engineering. *arXiv preprint arXiv:1606.02903*.
- Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., y Peterson, A. S. (1990). *Feature-oriented domain analysis (FODA) feasibility study*. Tech. rep. CMU/SEI-90-TR-21. Software Engineering Institute, Carnegie Mellon University.
- Kintz, M., Kochanowski, M., y Koetter, F. (2017). Creating User-specific Business Process Monitoring Dashboards with a Model-driven Approach. *Ponencia presentada en MODELSWARD. SCITEPRESS*, (pp. 353-361). doi: 10.5220/0006135203530361
- Kirda, E., y Kerer, C. (2000). MyXML: An XML based template engine for the generation of flexible Web content. *Ponencia presentada en WebNet World Conference on the WWW and Internet*, (pp. 317-322). Association for the Advancement of Computing in Education (AACE).
- Kuzilek, J., Hlosta, M., y Zdrahal, Z. (2017). Open university learning analytics dataset. *Scientific data*, 4(170171), 1-8. doi: 10.1038/sdata.2017.171.
- Liñán, L. C., y Pérez, Á. A. J. (2015). Educational Data Mining and Learning Analytics: differences, similarities, and time evolution. *International Journal of Educational Technology in Higher Education*, 12(3), 98-112. doi: 10.7238/rusc.v12i3.2515.
- Logre, I., Mosser, S., Collet, P., y Riveill, M. (2014). *Sensor data visualisation: a composition-based approach to support domain variability*. Ponencia presentada en European Conference on Modelling Foundations and Applications. 101-116. doi: 10.1007/978-3-319-09195-2_7.
- Long, P. D., y Siemens, G. (2011). Penetrating the Fog: Analytics in Learning and Education. *Educause Review*, 46(5), 30-32. doi: 10.17471/2499-4324/195.
- Magdaleníć, I., Radošević, D., y Kermek, D. (2011). Implementation Model of Source Code Generator. *Journal of Communications and Software Systems*. 7(2). 71-79. Doi: 10.24138/jcomss.v7i2.180.
- Mandinach, E. B., y Honey, M. (2008). *Data-driven school improvement: Linking data and learning*. Teachers College Press.
- Metzger, A., y Pohl, K. (2007). Variability management in software product line engineering. *Ponencia presentada en the Companion to the proceedings of the 29th International Conference on Software Engineering*, (pp. 186-187). doi: 10.1109/ICSECOMPANION.2007.83.
- Munzner, T. (2014). *Visualization analysis and design*. AK Peters/CRC Press.
- Palpanas, T., Chowdhary, P., Mihaila, G., y Pinel, F. (2007). Integrated model-driven dashboard development. *Information Systems Frontiers*, 9(2-3), 195-208. doi: 10.1007/s10796-007-9032-9.
- Patil, D., y Mason, H. (2015). *Data Driven: "O'Reilly Media, Inc."*. ISBN: 9781491925454.
- Picciano, A. G. (2012). The evolution of big data and learning analytics in American higher education. *Journal of Asynchronous Learning Networks*, 16(3), 9-20. doi: 10.24059/olj.v16i3.267
- Pleuss, A., Wollny, S., y Botterweck, G. (2013). Model-driven development and evolution of customized user interfaces. *Ponencia presentada en Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, (pp. 13-22). doi: 10.1145/2494603.2480298.

- Pohl, K., Böckle, G. y Van der Linden, F. J. (2005). *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc.
- Roberts, L. D., Howell, J. A., y Seaman, K. (2017). Give me a customizable dashboard: Personalized learning analytics dashboards in higher education. *Technology, Knowledge and Learning*, 22(3), 317-333. doi: 10.1007/s10758-017-9316-1.
- Ronacher, A. (2008). *Jinja2 Documentation*. Recuperado de <http://jinja.pocoo.org/docs/2.10/>:
- Sarikaya, A., Correll, M., Bartram, L., Tory, M., y Fisher, D. (2018). What Do We Talk About When We Talk About Dashboards? *IEEE Transactions on Visualization Computer Graphics*, 25(1), 682 - 692. doi: 10.1109/TVCG.2018.2864903.
- Schwendimann, B. A., Rodríguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., . . . Dillenbourg, P. (2017). Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies*, 10(1), 30-41. doi: 10.1109/TLT.2016.2599522
- Tajali, S. B., Corriveau, J.-P., y Shi, W. (2013). A Template-Based Approach to Modeling Variability. *Paper presented at the Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*.
- Tatsubori, M., y Suzumura, T. (2009). HTML templates that fly: a template engine approach to automated offloading from server to client. *Actas de 18th international conference on World wide web*, (pp. 951-960). doi: 10.1145/1526709.1526837.
- Teasley, S. D. (2017). Student facing dashboards: One size fits all? *Technology, Knowledge and Learning*, 22(3), 377-384. doi: 10.1007/s10758-017-9314-3.
- Van Gurp, J., Bosch, J., y Svahnberg, M. (2001). On the notion of variability in software product lines. Ponencia presentada en Software Architecture, 2001. *Proceedings. Working IEEE/IFIP Conference on Software Architecture*, Amsterdam, Netherlands, (pp. 45-54). doi: 10.1109/WICSA.2001.948406.
- Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., y Therón, R. (2019). *Dashboard Meta-Model for Knowledge Management in Technological Ecosystem: A Case Study in Healthcare*. Ponencia presentada en the UCAMi 2019, Toledo, Castilla-La Mancha, Spain. *MDPI Proceedings 2019*, 31(1), 44, 1-13. doi: 10.3390/proceedings2019031044.
- Vázquez-Ingelmo, A., García-Peñalvo, F. J., y Therón, R. (2019a). Addressing Fine-Grained Variability in User-Centered Software Product Lines: A Case Study on Dashboards. *Ponencia presentada en World Conference on Information Systems and Technologies. Advances in Intelligent Systems and Computing*, vol 930, (pp. 855-864). doi: 10.1007/978-3-030-16181-1_80.
- Vázquez-Ingelmo, A., García-Peñalvo, F. J., y Therón, R. (2019b). Capturing high-level requirements of information dashboards' components through meta-modeling. *Ponencia presentada en 7th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2019)*, León, Spain, 815-821, ACM. doi: 10.1145/3362789.3362837.
- Vázquez-Ingelmo, A., García-Peñalvo, F. J., y Therón, R. (2019c). Information Dashboards and Tailoring Capabilities - A Systematic Literature Review. *IEEE Access*, 7, 109673-109688. doi: 10.1109/ACCESS.2019.2933472.
- Vázquez-Ingelmo, A., García-Peñalvo, F. J., Therón, R., y Conde González, M. Á. (2019). Extending a dashboard meta-model to account for users' characteristics and goals for enhancing personalization. *Ponencia presentada en Learning Analytics Summer Institute (LASI) Spain*

2019, Vigo, Spain. CEUR Workshop Proceedings Series, no. 2415, (pp. 35-42). Recuperado de <http://ceur-ws.org/Vol-2415/papero4.pdf>

Yoo, Y., Lee, H., Jo, I.-H., y Park, Y. (2015). Educational dashboards for smart learning: Review of case studies. *Emerging issues in smart learning* (pp. 145-155): Springer.

PERFIL ACADÉMICO Y PROFESIONAL DE LOS AUTORES

Andrea Vázquez-Ingelmo. Investigadora en el grupo de investigación GRIAL. Graduada en Ingeniería Informática (2016, Universidad de Salamanca, España) y Máster en Ingeniería Informática (2018) por la misma universidad. Sus líneas de trabajo se centran en la visualización de datos y la ingeniería del software. ORCID: <https://orcid.org/0000-0002-7284-5593>
E-mail: andreavazquez@usal.es

Roberto Therón. Profesor Titular de Universidad del Departamento de Informática y Automática en la Universidad de Salamanca (USAL). Es encargado del grupo VisUsal (dentro del Grupo de Investigación GRIAL), el cual se centra en la combinación de enfoques procedentes de la Informática, Estadística, Diseño Gráfico y Visualización de Información, para obtener una adecuada comprensión de conjuntos de datos complejos. En los últimos años, se ha dedicado al desarrollo de herramientas de visualización avanzada para datos multidimensionales, como por ejemplo datos genéticos o paeloclimáticos. En el área de Analítica Visual desarrolla productivas colaboraciones con grupos e instituciones de reconocido prestigio internacional, como el Laboratorio de Ciencias del Clima y del Medio Ambiente (París) o el Centro de Analítica Visual Avanzada de la ONU (Suiza). ORCID: <https://orcid.org/0000-0001-6739-8875>
E-mail: theron@usal.es

Dirección:
Facultad de Ciencias
Universidad de Salamanca
Plaza de los caídos, s/n
Salamanca (España)

Fecha de recepción del artículo: 14/01/2020
Fecha de aceptación del artículo: 02/03/2020
Fecha de aprobación para maquetación: 15/03/2020