



Revista de Investigación en Tecnologías de la Información

ISSN: 2387-0893

editorial@riti.es

Universitat Politècnica de Catalunya

España

Martínez Martínez, Víctor Alfonso; Ramírez, Lorena Alonso; Montané Jiménez, Luis Gerardo

**Learning analytics dashboard para apoyar la enseñanza
de la programación en educación a distancia**

Revista de Investigación en Tecnologías de la Información,
vol. 12, núm. 26, 2024, julio-diciembre, pp. 164-178

Universitat Politècnica de Catalunya

España

DOI: <https://doi.org/10.36825/RITI.12.26.013>

- Número completo
- Más información del artículo
- Página de la revista en portal.amelica.org





Learning analytics dashboard para apoyar la enseñanza de la programación en educación a distancia

Learning analytics dashboard to support programming education in distance learning

Víctor Alfonso Martínez Martínez

Universidad Veracruzana, Xalapa, México

zs22000730@estudiantes.uv.mx

Lorena Alonso Ramírez

Universidad Veracruzana, Xalapa, México

lalonso@uv.mx

Luis Gerardo Montané Jiménez

Universidad Veracruzana, Xalapa, México

lmontane@uv.mx

doi: <https://doi.org/10.36825/RITI.12.26.013>

Recibido: Junio 20, 2024

Aceptado: Diciembre 12, 2024

Resumen: Al enseñar a programar en una modalidad educativa mixta o a distancia, las interacciones presenciales que orientan a los docentes en la identificación de aptitudes o problemas en la materia disminuyen, incluso en algunos casos son inexistentes, por lo cual el aprendizaje de los alumnos puede resultar deficiente. El objetivo de este estudio es apoyar a dichos docentes a identificar a aquellos alumnos que presentan dificultades con la materia. A través de un proceso de *Learning Analytics* se definieron y se crearon los mecanismos para recopilar métricas de desempeño de los alumnos, que a través de técnicas de visualización de datos son plasmadas en un *dashboard* con el fin de aminorar esta brecha de interacción. Para efecto de la investigación, se realizó un diseño cuasi experimental aplicando la técnica de entrevista con el fin de obtener datos cualitativos. Los resultados obtenidos muestran la efectividad de la herramienta al poder identificar a los estudiantes con problemas en la materia, además de los tópicos que deben ser reforzados de manera general.

Palabras clave: *Analíticas del Aprendizaje, Visualización de Datos, Panel Analítico, Enseñanza de Programación.*

Abstract: When teaching programming in a blended or distance learning mode, the face to face interactions that help teachers identify students' aptitudes or problems in the subject decrease or, in some cases, may be non-existent. As a result, students' learning may be deficient. The objective of this study is to support teachers in identifying students who are struggling with the subject. Therefore, through a Learning Analytics process, mechanisms were defined and created to collect student performance metrics. These metrics are displayed on a dashboard using Data Visualization techniques to bridge this interaction gap. For the research, a quasi-experimental design was implemented, applying the interview technique to obtain qualitative data. The results

show the effectiveness of the tool in identifying students who are having difficulties with the subject, as well as the topics that need to be reinforced generally.

Keywords: *Learning Analytics, Data Visualizations, Analytics Dashboard, Programming Teaching.*

1. Introducción

Enseñar a programar es un proceso complejo, consistente en plasmar la solución a un problema mediante un lenguaje de programación específico. Por lo tanto, no es solo enseñar un procedimiento o fórmula para que se aplique repetidas veces. Cada problema puede resolverse en más de una forma y cada programador elige una de ellas, es ahí donde recae la dificultad de enseñar a programar [1].

Al enseñar a programar dentro de un aula, las interacciones cara a cara, la finalización de tareas o la consecución de objetivos por parte de los alumnos le facilitan al docente la distinción de aquellos estudiantes que muestran una mayor o menor comprensión en los temas de la materia. Con base en estos datos, el docente puede decidir, entre otras cosas, el reforzamiento de puntos específicos o apoyar a los alumnos que así lo necesiten.

Cuando los estudiantes son trasladados a un entorno distribuido mediante alguna plataforma educativa, las interacciones mencionadas anteriormente resultan mermadas. Después de la revisión de diversos trabajos enfocados en la enseñanza de la programación en modalidades mixta o a distancia (Ghorashi y Jensen [2]; Langton *et al.* [3], por citar algunos) y del análisis del contexto en el que estos docentes se desenvuelven, se identifica que una de estas mermas es la falta de mecanismos apropiados que muestren la información sobre el desempeño individual de los estudiantes. Esta falta de información puede resultar en que el docente no conozca —entre otras cosas— la participación de cada estudiante o los errores cometidos al realizar algún ejercicio. Como resultado, podría identificarse una disminución de los aprendizajes esperados o elevar el índice de reprobación entre los alumnos.

Con el fin de aminorar la brecha de interacción entre la educación presencial y la educación a distancia, se utilizó un proceso de *Learning Analytics* (LA), el cual se define como la medición, estructuración, análisis y el reporte de datos sobre los alumnos y sus contextos de aprendizaje. Su objetivo es mejorar el aprendizaje y el entorno en el que este se desarrolla, además de proporcionar a los involucrados (docentes, tutores, padres y estudiantes) una retroalimentación adecuada y rápida sobre su aprendizaje [4].

Los resultados obtenidos se presentan al docente utilizando técnicas de visualización de datos, la cual es la representación de información mediante elementos gráficos, por ejemplo, mapas o gráficas estadísticas. De acuerdo con distintos estudios (Islam y Jin [5]; Valero [6]) esto agiliza la interpretación de la información por parte del cerebro humano, pues por naturaleza, para las personas es más sencillo interpretar información cuando esta se presenta en forma gráfica que cuando se muestra en números o textos.

Las técnicas de visualización elegidas se implementarán en un *dashboard* pues es una interfaz visual que muestra de manera simplificada la información relevante para alcanzar uno o varios objetivos. Su diseño permite un monitoreo rápido y efectivo con solo un vistazo [7].

El presente documento detalla el proceso seguido para la recolección y procesamiento de las métricas de desempeño asociadas a un grupo de alumnos de programación básica, así como la evaluación de un primer prototipo funcional del *dashboard* en la que estas le serán presentadas al docente a cargo, todo esto correspondiente a una primera iteración en un modelo de *Learning Analytics*.

2. Estado del arte

En el marco de la presente investigación se implementó una Revisión Sistemática de Literatura (RSL) basada en la metodología de Kitchenham [8]. Se buscó responder tres preguntas para apoyar el desarrollo práctico del proyecto, y por supuesto, encontrar los trabajos relacionados a este. Las preguntas formuladas son las siguientes:

1. ¿Qué medios se utilizan para recolectar indicadores de desempeño de aprendizaje?
2. ¿Qué indicadores pueden utilizarse para medir el desempeño de los alumnos?
3. ¿Cómo se aplican los resultados obtenidos?

Una vez aplicados cada uno de los pasos de la metodología se obtuvieron las siguientes respuestas para las preguntas de investigación. Respecto a la pregunta número uno, en la Tabla 1 se muestra los medios con los cuales se recolectan datos para evaluar el desempeño de los estudiantes.

Tabla 1. Medios para recolectar datos sobre el aprendizaje de los estudiantes.

Medio	Descripción
Recolección directa	La información se obtiene mediante la aplicación de evaluaciones o cuestionarios a los estudiantes.
Interacciones sociales	Se obtiene al medir la participación de los estudiantes en blogs, foros, redes sociales o chats. En ocasiones estos se encuentran embebidos en las propias plataformas de aprendizaje.
Uso de recursos de ayuda	Se monitorea el uso de documentación de herramientas o búsquedas en internet que ayuden a los estudiantes a resolver ejercicios.
Interacción con el software	Recopila datos acerca de las interacciones de los estudiantes al momento de operar una herramienta de software, comúnmente sin que este lo perciba.
Sensores físicos	Recopila información de los estudiantes utilizando herramientas de hardware como, cámaras, micrófonos, dispositivos de seguimiento ocular, monitores de ritmo cardíaco y sensores de presión.
Información histórica	Datos del estudiante almacenados en los sistemas escolares.

Fuente: Elaboración propia a partir de los trabajos de investigación obtenidos en la RSL.

Cada medio de recolección de información provee a los interesados distintos indicadores de desempeño de los alumnos. Estos indicadores pueden ser empleados en distintas áreas de conocimiento, en la Tabla 2 se muestran los indicadores asociados a cada uno de ellos, los cuales responden a la pregunta número dos.

Tabla 2. Indicadores de desempeño por medio de recolección.

Medio	Indicadores
Recolección intencional	• Respuestas cualitativas • Respuestas cuantitativas
Interacciones sociales	• Cortesía • Estrés • Acuerdos • Desacuerdos
Uso de recursos de ayuda	• Palabras de búsqueda • Secciones de documentación
Interacción con el software	• Pulsación de teclas • Operaciones con el portapapeles • Abrir • Cerrar • Guardar • Compilaciones • Errores • Tiempo • Resultados • Acciones del ratón
Sensores físicos	• Respuestas emocionales • Fijaciones • Sacadas • Ritmo cardíaco • Presión hacia el teclado o ratón
Información precargada	• Edad • Sexo

-
- Localidad
 - Calificaciones
 - Cursos (escolaridad)
-

Fuente: Elaboración propia a partir de los trabajos de investigación obtenidos en la RSL.

En este proyecto, los indicadores obtenidos a través de la “interacción con el software” son los que cobran vital importancia pues son la base de información para el *dashboard*.

Finalmente, en respuesta a la pregunta número tres se identifica que en un 45% de las investigaciones se emplean LADs como medio de presentación de la información, lo que remarca la importancia que este medio de visualización está cobrando en la actualidad.

2.1. Trabajos relacionados

Después de la revisión de artículos enfocados en *Learning Analytics*, se advierte que este aún es presentado como un campo emergente o en etapa de maduración; sin embargo, a la fecha existe una alta cantidad de documentos enfocados en el tema, lo cual evidencia su crecimiento y utilidad en el campo educativo.

Es en el campo de la enseñanza de programación, donde el LA presenta un crecimiento reducido, ofreciendo una cantidad menor de artículos disponibles con respecto a otras áreas. Dentro de este tipo de artículos, destacan trabajos dirigidos a evaluar el pensamiento computacional de los estudiantes para apoyar a los docentes. Trabajos como los de Grover *et al.* [9] y Ruiperez *et al.* [10] presentan una recolección automática de indicadores de desempeño a través de la interacción de los estudiantes con un entorno de aprendizaje, en este caso los ejercicios son de carácter lúdico, pues las investigaciones están dirigidas a alumnos preuniversitarios.

En la investigación propuesta por Hundhausen *et al.* [11] se sugiere el uso de *plug-ins* dentro de distintos IDEs con el fin de recolectar diversas métricas de desempeño. Una vez obtenidas los docentes desarrollarán diversas intervenciones, las cuales retornarán directamente a cada uno de los estudiantes.

En lo que respecta a la investigación de Olivares *et al.* [12], se propone adecuar un IDE para incluir la recolección de métricas de carácter social mediante las interacciones de los estudiantes en un chat del grupo y evaluar cómo influyen en la consecución de resultados.

Se puede distinguir que en cada una de las investigaciones se utiliza el entorno de desarrollo en el que interactúan los estudiantes como base para la recolección de información, sin embargo, tanto el nivel en el que estas se aplican como la presentación de resultados varían a lo pretendido en el presente proyecto, lo que conlleva a que este trabajo cubra un aspecto de los menos abordados en el área.

3. Materiales y métodos

A continuación, se describen las etapas para la elaboración del *dashboard*, que como se mencionó anteriormente, está construido mediante un proceso de *learning analytics*. Se tomó como base el modelo propuesto por Amo y Santiago [13] el cual consta de cuatro fases: 1) establecer objetivos y métricas, 2) recolectar datos, 3) analizar y visualizar información, 4) actuar (para fines del presente estudio esta etapa se omitirá, pues las acciones se dejan a consideración de los docentes), y 5) evaluación.

3.1. Establecer objetivos y métricas

En la presente investigación, se estableció como objetivo apoyar al docente que imparte cursos de programación básica en modalidades mixta o a distancia a identificar a aquellos alumnos que presentan dificultades con la materia, esto a través de la presentación de indicadores de desempeño de los alumnos en un *dashboard*.

Como paso inicial, se investigaron las distintas métricas que pudieran obtenerse automáticamente a partir de la resolución de ejercicios de programación por parte de los alumnos en un ambiente de desarrollo. De los distintos indicadores identificados en la revisión del estado del arte, se eligió obtener los siguientes:

1. Avance: la cantidad de ejercicios finalizados por el alumno.
2. Código: el código escrito por el alumno para resolver cada ejercicio.
3. Errores: número de errores que tuvo el alumno en cada ejercicio.
4. Intentos: número de veces que el alumno intentó resolver el ejercicio.
5. Resultado: resultado obtenido al ejecutar el código.

6. Tiempo: tiempo empleado para resolver cada ejercicio.
7. Tiempo de ejecución: tiempo en que se ejecuta el código de cada ejercicio.
8. Uso de memoria: memoria empleada al ejecutar el código de cada ejercicio.

3.2. Recolectar datos

Se llevó a cabo el desarrollo de una plataforma en la que los alumnos, una vez autenticados, puedan acceder a las materias de programación que tengan inscritas en el ciclo escolar. Cada una de ellas contará con los ejercicios que los profesores hayan registrado previamente y que serán resueltos y compilados dentro de esta.

El almacenamiento de información se lleva a cabo en una base de datos implementada en el sistema de gestión de bases de datos relacional MySQL, elegido principalmente por su amplia utilización en aplicaciones web, además de ser de uso gratuito.

Para efectos de compilación se utilizó JDoodle [14], la cual es una plataforma de programación en línea que cuenta con los siguientes servicios: compilador e IDE en línea para más de 76 lenguajes de programación y dos bases de datos, un API (interfaz de programación de aplicaciones, por sus siglas en inglés) de compilación basado en REST, un *plug-in* para incluir un IDE en aplicaciones web, así como una plataforma de cursos y evaluación en línea de programación.

En este caso se utilizó el servicio de API con el cual se obtiene una respuesta después de cada intento de compilación, los demás servicios fueron descartados pues no proporcionan la información requerida para la investigación.

Para obtener y almacenar la totalidad de los indicadores seleccionados se utilizaron elementos de interacción de la plataforma desarrollada, es decir, cada vez que una acción involucre la generación de uno de los indicadores descritos, se dispara un evento que crea o actualiza la información del alumno en tiempo real. En la Figura 1 se muestra un ejemplo de la interfaz del entorno de desarrollo.



Figura 1. Interfaz del entorno de desarrollo diseñado para la recolección de métricas. La sección izquierda muestra las instrucciones del ejercicio y el botón Iniciar (cambia su etiqueta a Finalizar automáticamente); en la parte central se despliega la caja de escritura de código, la sección de resultados y el botón de compilar; finalmente, la sección derecha presenta una caja de texto para la introducción de variables si es que el ejercicio solicita entradas desde la interfaz.

La Tabla 3 describe cómo se obtiene cada indicador y el evento que dispara su almacenamiento en la base de datos, por cada uno de los alumnos y ejercicio de manera individual.

Tabla 3. Obtención de las métricas.

Métrica	Obtención	Evento
Avance	Conteo de los ejercicios marcados como finalizados por cada uno de los alumnos del grupo.	Al dar clic en el botón Finalizar ejercicio, este se marca como entregado y se contabiliza en el avance del alumno.
Código	Directamente del área de escritura del código.	Al dar clic en el botón Compilar se inserta o actualiza el código del ejercicio.
Reintentos por errores en el código	Obtenido a través del parámetro de retorno <i>output</i> del API.	Al recibir respuesta del API después de dar clic al botón Compilar, se evalúa el parámetro de retorno <i>output</i> ; si contiene un mensaje de error se suma un error al ejercicio del alumno.
Intentos	Conteo de veces que el alumno ingresa para resolver un ejercicio específico.	Al dar clic en el botón Iniciar se suma uno al número de intentos.
Resultado	Directamente del área de resultado de la compilación.	Al dar clic en el botón Compilar, se inserta o actualiza el resultado del ejercicio.
Tiempo	Tiempo empleado entre el inicio de resolución del ejercicio y la marcación de este como finalizado.	Al dar clic en el botón Iniciar, se crea el registro del ejercicio incluyendo la hora de inicio. Al dar clic en el botón Finalizar, se actualiza el registro con la hora de finalización. El tiempo para resolver el ejercicio se obtiene restando la hora inicial de la final.
Tiempo de ejecución	Tiempo empleado para ejecutar el código.	Al recibir respuesta del API después de dar clic al botón Compilar, se almacena el parámetro de retorno <i>cpuTime</i> .
Uso de memoria	Memoria utilizada al ejecutar el código.	Al recibir respuesta del API después de dar clic al botón Compilar, se almacena el parámetro de retorno <i>memory</i> .

Fuente: Elaboración propia.

3.3. Analizar y visualizar información

Como se mencionó anteriormente, cada uno de los indicadores se registra por alumno y por ejercicio, por lo cual es necesario generar resúmenes que concentren el desempeño general del grupo de manera simplificada. Para ello, se crearon distintas tablas en la base de datos que serán alimentadas mediante el uso de procedimientos almacenados, los cuales se ejecutan al momento en el que el profesor da clic en el botón *estadísticas* del grupo que desea evaluar (mostrado en la Figura 2), con el fin de que cuente con la información totalmente actualizada.

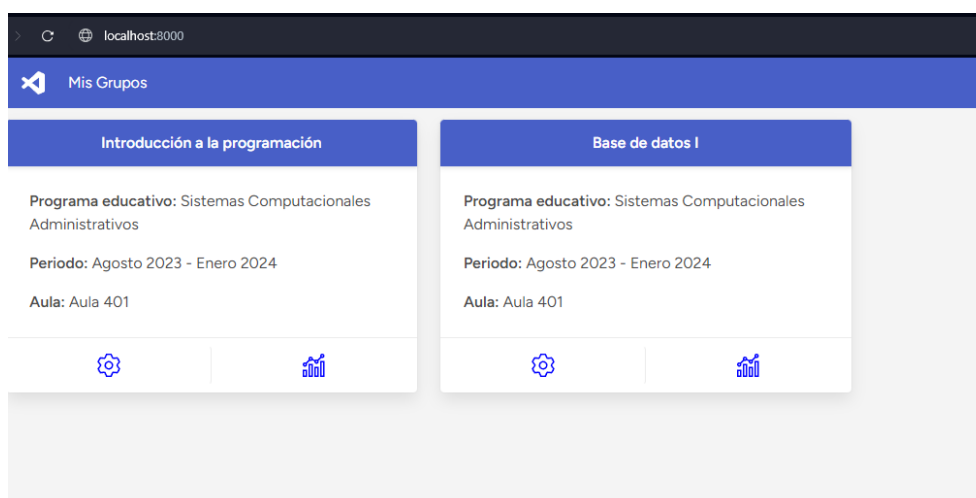


Figura 2. Cartas de selección de los grupos que tiene asignados un profesor, muestra los botones de configuración y de estadísticas para cada una de sus materias.

Los datos de las tablas de resumen servirán como base para los gráficos mostrados en el *dashboard*. Es importante señalar que el diseño de este se basó en la metodología de Diseño Centrado en el Usuario (DCU).

El estándar internacional ISO 9241 apartado 210 Diseño Centrado en el Humano para Sistemas Interactivos [15], nos dice que un diseño centrado en el humano mejora la eficacia y la eficiencia, así como el bienestar humano y su satisfacción, la accesibilidad y la sustentabilidad. Además, contrarresta posibles efectos adversos sobre la salud humana, la seguridad y el rendimiento.

Este estándar propone además un modelo de cuatro etapas iterativas, las cuales indica, se pueden integrar a cualquier metodología de desarrollo. A continuación se da una breve descripción de cada una de ellas.

Entendimiento y especificación del contexto de uso. Se basa en la especificación de los siguientes puntos:

- Los usuarios e interesados: deben identificarse claramente los grupos y describir su relación con el desarrollo en términos de metas y limitaciones.
- Las características de los usuarios o grupos de usuarios: se identificarán las características pertinentes de los usuarios, pueden incluir conocimientos, habilidades, experiencia, educación, capacitación, atributos físicos, hábitos, preferencias y capacidades.
- Los objetivos y tareas de los usuarios: se identificarán los objetivos de los usuarios y el objetivo general del sistema.
- El entorno del sistema: debe identificarse el entorno técnico, incluidos el hardware, el software y los materiales. Además, las características relevantes del medio físico, social y cultural.

Estos puntos deben describirse con suficiente detalle pues respaldarán los requisitos, el diseño y las actividades de evaluación.

Especificación de los requerimientos del usuario. Esta actividad es común a cualquier desarrollo de sistemas —identificar los requerimientos del usuario y especificar los requisitos funcionales—. Para el diseño centrado en el usuario, esta actividad debe ampliarse con los requisitos del usuario obtenidos en la fase anterior.

Producción de soluciones de diseño. Generar un diseño que satisfaga la experiencia del usuario es un proceso de innovación que no solo se basa en la eficacia y la eficiencia del sistema, sino que se toman en cuenta los aspectos emocionales del usuario. Al igual que la fase anterior, se toma como base la información recolectada en el contexto de uso además de la experiencia proporcionada por los miembros de un equipo multidisciplinario. En esta etapa deben considerarse los siguientes puntos:

- Diseñar la interacción usuario-sistema mediante una interfaz apropiada que cumpla con los requisitos de este.
- Concretar las soluciones de diseño.
- Modificar las soluciones de diseño en respuesta a la evaluación y retroalimentación del usuario.
- Comunicar las soluciones a los responsables de su desarrollo/implementación.

Evaluación del diseño. Se debe evaluar el diseño desde etapas tempranas, ya que mientras más se avance en el proyecto, las modificaciones podrían resultar más costosas. Es importante remarcar que el DCU es un proceso

iterativo como se esquematiza en la Figura 3, por lo que se puede ir de una fase a otra las veces que sean necesarias con tal de asegurar la satisfacción del usuario.



Figura 3. Etapas del DCU propuestas por el estándar ISO 9242. Imagen tomada de <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/human-centered-design/>

En los siguientes párrafos, se resumen las acciones tomadas para cubrir, lo correspondientes a cada una de las etapas del DCU.

En las **etapas de entendimiento del contexto y definición de actividades** se implementó el *framework* PACT (Personas, Actividades, Contexto y Tecnologías). La premisa de este es analizar a las personas y sus actividades dentro de un contexto específico, además de identificar la tecnología utilizada en dichas actividades.

Los resultados para cada apartado del análisis PACT fueron los siguientes, es importante señalar que en este caso se contó en todo momento con la participación de un usuario dedicado a la docencia en línea, así como con profesores de la Facultad de Estadística e Informática de la Universidad Veracruzana para complementar los requerimientos:

Personas: Quiénes son los usuarios, sus características y necesidades, para ello se implementó una ficha persona, el cual es un perfil que representa a tu usuario o usuarios objetivo. Se basa en los datos recabados de los usuarios reales y sus comportamientos, necesidades, y objetivos. Para este se realizó una ficha persona de un profesor universitario que imparte clases de programación básica.

Actividades: Qué tareas realizan los usuarios y cómo interactúan con el sistema.

Actividades generales:

- Planificar e impartir clases tanto teóricas como prácticas a los estudiantes.
- Desarrollar materiales como presentaciones, ejercicios y recursos educativos para apoyar la enseñanza.
- Supervisar y asesorar a los estudiantes en proyectos de investigación, tesis o proyectos de la carrera.
- Elaborar, aplicar y calificar exámenes, proyectos y tareas, así como proporcionar retroalimentación a los estudiantes.
- Mantenerse actualizado en el campo de la informática a través de la investigación continua.
- Ofrecer orientación académica y profesional a los estudiantes, brindando asesoramiento sobre la carrera a través de los programas de tutorías.
- Mantener contacto con el sector laboral para facilitar oportunidades de prácticas y servicios sociales para los estudiantes, además de supervisar su progreso.
- Contribuir a la investigación académica por medio de la publicación de artículos en revistas académicas, conferencias y foros.

Actividades en línea:

- Revisar y cargar a la plataforma los materiales educativos que utilizará durante el curso.
- Impartir sesiones mediante videoconferencias.
- Revisar cada una de las tareas de los estudiantes y retroalimentarlas.
- Responder a las preguntas y dudas que los estudiantes planteen a través del correo electrónico o mensajes dentro de la plataforma.
- Preparar los exámenes del curso y subirlos a la plataforma.
- Calificar los exámenes y actualizar las calificaciones en los periodos establecidos.

Contexto: El entorno en el que se utilizan los productos, incluyendo factores físicos y sociales.

- **Contexto físico:** Por la naturaleza del modelo de educación a distancia el entorno es altamente variable, algunos profesores pueden estar en entornos silenciosos y aislados, mientras que otros enfrentan distracciones en medios más ruidosos o compartidos. La iluminación puede ser adecuada para estar largos periodos frente a un monitor de computadora y en otros casos puede carecer de luz adecuada o reflejos en la pantalla por exposición a la luz solar. Esta variabilidad puede afectar la atención y la participación del docente, así como la de los estudiantes.
- **Contexto social:** Este contexto es el más afectado en la educación a distancia, pues en ella se pierden las interacciones que brinda la educación presencial. Las prácticas sociales se limitan a aquellas que permitan las plataformas utilizadas, ya sea videoconferencias, chat grupal o foros, usualmente dando retroalimentación a las actividades de forma textual de forma asíncrona.

Tecnología: Las herramientas y dispositivos que los usuarios utilizan, en esta modalidad de estudio es fundamental, pues abarca las herramientas y plataformas tecnológicas utilizadas para facilitar la enseñanza.

- **Plataformas:** seleccionar una plataforma de enseñanza en línea es fundamental para mantener una buena experiencia de enseñanza-aprendizaje. Se puede optar por incluir sistemas de gestión de aprendizaje como Moodle o Canvas, que proporcionan un entorno centralizado para cargar materiales de curso, interactuar con estudiantes y realizar evaluaciones.
- **Herramientas de videoconferencia:** la elección de herramientas de videoconferencia, como Zoom, Microsoft Teams o Google Meet, impacta directamente en la calidad de las sesiones en línea. Es esencial considerar la estabilidad de la conexión, las capacidades de interacción en tiempo real y las opciones de grabación para que, dado el caso, se pueda hacer una revisión de las clases.
- **Recursos digitales:** La selección y creación de recursos digitales, como presentaciones, videos y actividades, contribuye significativamente a la experiencia de enseñanza-aprendizaje en línea.
- **Comunicación:** además de las herramientas de videoconferencia, las plataformas de correo, mensajería y foros en línea facilitan la comunicación, permitiendo la interacción fuera de las horas de clase en tiempo real.
- **Infraestructura:** se debe tener presente que cada uno de los individuos involucrados cuenta con distintos equipos de cómputo y conexión a internet, lo cual influirá en la calidad de las sesiones programadas.

Con base en los resultados de las etapas previas, se abordó la **etapa de diseño**. Es importante señalar que, a partir del análisis con el usuario, en el *dashboard* se visualizan únicamente las métricas avance, errores y tiempo, además de una referencia a los ejercicios calendarizados. Las métricas restantes se descartaron de la vista principal, pues se indicó que son referentes para alumnos con un nivel más avanzado de conocimientos; sin embargo, se utilizaron para dar un detalle específico por ejercicio/alumno.

En esta etapa se produjeron *wireframes* (Figura 4) y prototipos de baja fidelidad (Figura 5) los cuales fueron retroalimentados por un usuario.

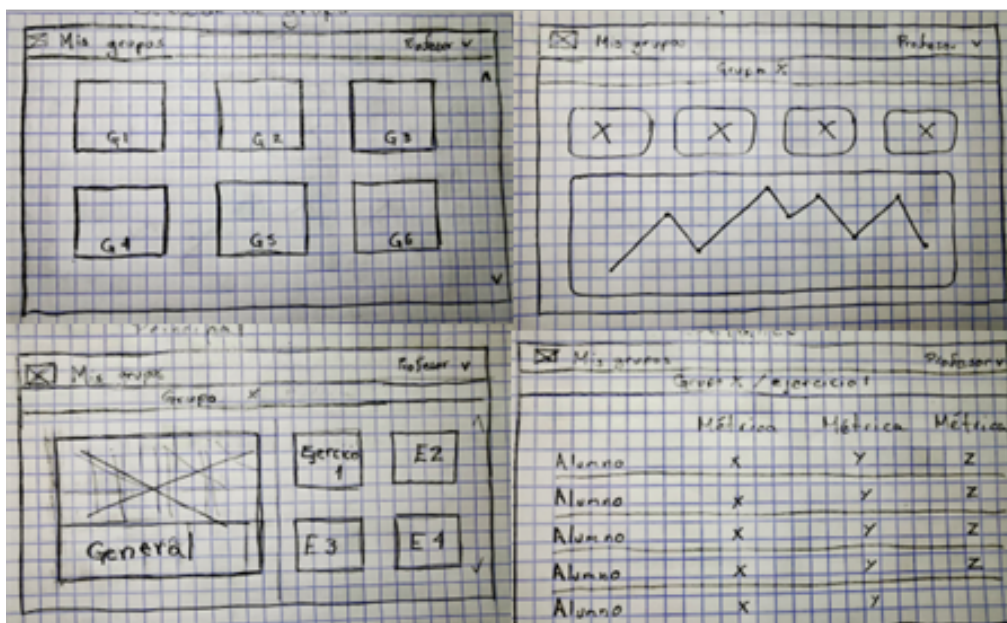


Figura 4. Ejemplos de wireframes desarrollados para el diseño del *dashboard*.

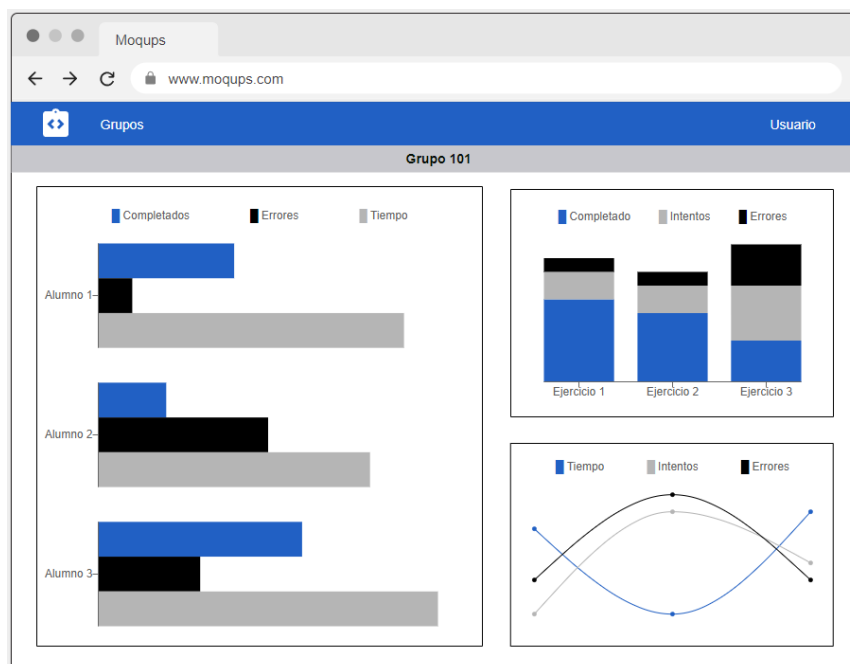


Figura 5. Prototipo de baja fidelidad utilizado para el desarrollo del *dashboard* creado en la plataforma moqups.com.

Después de iterar en distintas ocasiones con los *mockups* se llegó a una versión final la cual fue trasladada a un prototipo de alta fidelidad utilizando la plataforma *Looker Studio* de Google (Figura 6). Esta versión fue evaluada por tres usuarios distintos, utilizando la técnica de entrevista en la cual se buscó obtener resultados de tipo cualitativo. En esta evaluación se recibieron calificaciones positivas y un mínimo de cambios en el apartado gráfico por lo que no hubo necesidad de realizar nuevas iteraciones para trasladarlo a un prototipo funcional.

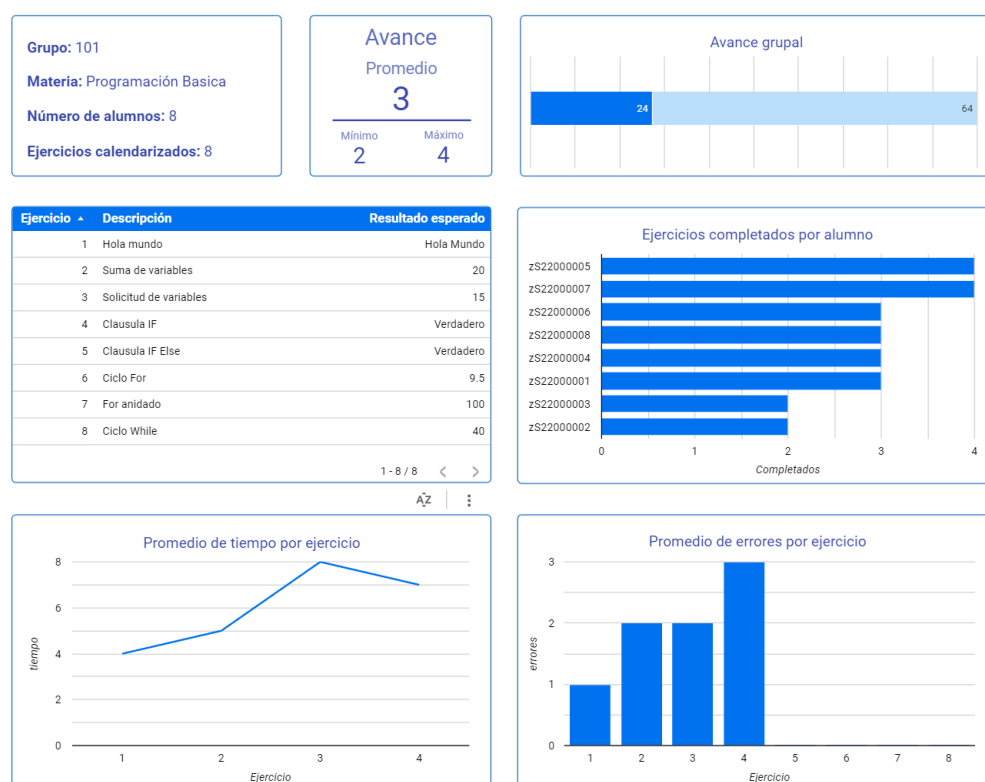


Figura 6. Prototipo de alta fidelidad diseñado en *Looker Studio* de Google.

3.4. Evaluación

Se llevó a cabo una evaluación con cinco participantes, cuya selección obedece a un diseño cuasi-experimental, es decir, no se eligieron de forma aleatoria. El número de participantes fue elegido basándose en la regla propuesta por Nielsen [16], la cual dicta que probar con cinco usuarios es suficiente para identificar la mayoría de los problemas de usabilidad de una interfaz o producto digital. Como se detalla en Verbert *et al.* [17], se llevó a cabo una prueba de utilidad, estas pruebas son las que se realizan con mayor frecuencia, ya sea con profesores, estudiantes o ambos, se enfocan principalmente en plantear actividades a los docentes, relacionadas a encontrar alumnos en riesgo o preguntarles qué tan bien creen que se están desempeñando en un curso.

Con respecto al diseño y satisfacción del usuario, se hizo uso del cuestionario CSUQ. Se eligió este cuestionario ya que como se reporta en Hedlefs *et al.* [18], en un estudio comparativo entre diferentes instrumentos de evaluación de usabilidad, se encontró que el CSUQ tiene un buen desempeño midiendo las reacciones de los participantes a un sitio web, en comparación con otros instrumentos, como son: Words, QUIS y SUS. Además, de que realizaba evaluaciones válidas aun con muestras pequeñas.

El prototipo funcional del *dashboard* empleado para la prueba se presenta en la Figura 7. Como puede observarse cuenta con los mismos elementos del prototipo de alta fidelidad con un cambio en la distribución de la información, pues gracias al proceso de DCU las adecuaciones en la implementación son mínimos.

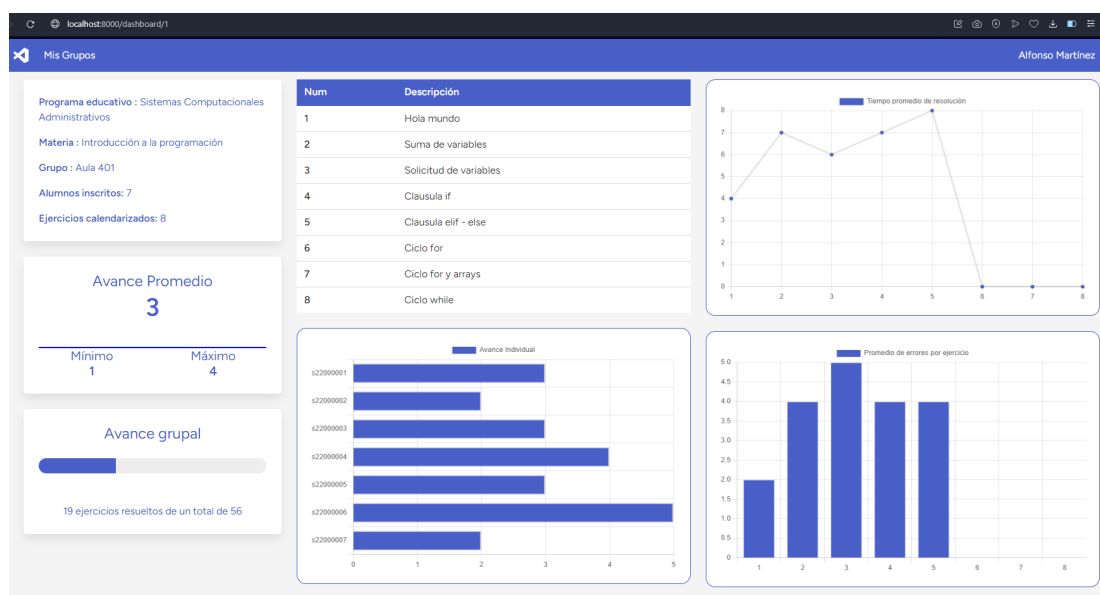


Figura 7. Prototipo funcional del *dashboard* con el que se llevaron a cabo las pruebas de utilidad.

En este prototipo se puede llegar a un segundo nivel de detalle al dar clic en la fila del ejercicio que el profesor desee revisar. Como se mencionó en párrafos anteriores dicho detalle se visualiza por ejercicio/alumno en una tabla simple donde se muestran las demás métricas recolectadas (Figura 8), lo cual sirve para tener un panorama más preciso de los alumnos y como solucionaron los ejercicios, incluso sirviendo como referencia para asentar calificaciones.

Mis Grupos								
Suma de variables								
Matrícula	Nombre	Tiempo	Intentos	Errores	Memoria (Kb)	Tiempo CPU (seg)	Resultado	Código
zs23000008	Victor Martinez	00:06:00	1	3	6136	0.03	20	20
zs23000001	Abigail Artigas	00:06:00	3	2	5594	0.06	20	20
zs23000002	Pablo Rivera	00:05:00	2	1	6076	0.01	20	20
zs23000003	Armando Alarcón	00:07:00	1	1	6792	0.05	20	20
zs23000004	Abril González	00:06:00	3	2	5736	0.02	20	20
zs23000005	Elisa García	00:07:00	3	3	6204	0.05	20	20
zs23000006	Angélica Ameca	00:05:00	1	3	5700	0.07	20	20
zs23000007	Carol Medina	00:05:00	1	1	6416	0.01	20	20

Figura 8. Tabla de detalle por ejercicio/alumno.

4. Resultados

Como se mencionó anteriormente, se procuró obtener resultados de utilidad, satisfacción y diseño, estos se detallan a continuación.

4.1. Utilidad

Este punto fue medido basándose en la lista de tareas (ver Figura 9) que se le solicitó realizar a los usuarios dentro de la plataforma, se utilizaron dos parámetros como punto de referencia, el tiempo de resolución, y la comparación de las respuestas obtenidas contra la plantilla de respuestas esperadas.

Hoja de actividades

Evaluación de usabilidad/utilidad de un *Learning Analytics Dashboard* que apoye la enseñanza de programación básica en ambientes mixtos o a distancia.

- Acceda a la plataforma mediante el correo y contraseña proporcionados por el facilitador.
- Tome unos segundos para identificar los grupos que tiene asignados y después elija la opción “estadísticas” de la materia “Introducción a la programación” de la Licenciatura en Ingeniería del Software.
- Una vez que se encuentre en la sección estadística del grupo y de acuerdo con la información mostrada en los gráficos responda los siguientes puntos:
 - Escriba la matrícula de 3 alumnos que necesitarían un mayor apoyo en la clase.
 - Escriba la matrícula de 2 alumnos que podrían asesorar a los alumnos con rezago.
 - Escriba el nombre del tema o temas que crea debería reforzar en clases.
- Acceda a información detallada del ejercicio número 3 y conteste los siguientes planteamientos.
 - Escriba la matrícula de 2 alumnos que no terminaron el ejercicio.
 - Seleccione el código de un alumno al azar y escriba si se visualiza de forma correcta.
- Regrese a la sección “Mis Grupos” y acceda a la sección de configuración de cualquiera de las materias.

Gracias por su participación.

Figura 9. Hoja de actividades.

El tiempo de resolución de las actividades oscilo entre los 6:07 y los 8:02 minutos, para identificar lo solicitado dentro de un grupo hipotético de ocho alumnos registrados con igual número de ejercicios calendarizados. Los tiempos obtenidos resultaron satisfactorios pues se estimó un total de 7 minutos.

De las cuatro actividades en las que se solicitaba escribir una respuesta, se realizó un *check list* de verificación, evaluando las respuestas obtenidas contra la plantilla de respuestas esperadas, se obtuvo un 70% de concordancia exacta, un 20% de concordancia parcial y un 10% de error, cantidades que sugieren que mediante el *dashboard* es posible identificar problemas generales dentro del curso, así como alumnos en riesgo o aquellos que tienen un mejor desempeño.

4.2. Satisfacción y diseño

Como se mencionó previamente la satisfacción de los usuarios se midió utilizando el cuestionario CSUQ (ver Figura 10), mediante este se evalúan las siguientes dimensiones, calidad del sistema (satisfacción general), calidad de la información y calidad de la interfaz y diseño. Su interpretación consiste en un sistema de puntuación en escalas de Likert donde los valores más cercanos a 7 indican una mayor satisfacción o percepción positiva. Es importante señalar que la versión en español del cuestionario se encuentra validada en la publicación de Hedlefs *et al.* [18].

	Totalmente en desacuerdo				Totalmente de acuerdo		
	1	2	3	4	5	6	7
1 En general, estoy satisfecho con lo fácil que es utilizar este sitio web.	☐	☐	☐	☐	☐	☐	☐
2 Fue simple usar este sitio web.	☐	☐	☐	☐	☐	☐	☐
3 Soy capaz de completar mi trabajo rápidamente utilizando este sitio web.	☐	☐	☐	☐	☐	☐	☐
4 Me siento cómodo utilizando este sitio web.	☐	☐	☐	☐	☐	☐	☐
5 Fue fácil aprender a utilizar este sitio web.	☐	☐	☐	☐	☐	☐	☐
6 Creo que me volví experto rápidamente utilizando este sitio web.	☐	☐	☐	☐	☐	☐	☐
7 El sitio web muestra mensajes de error que me dicen claramente cómo resolver los problemas.	☐	☐	☐	☐	☐	☐	☐
8 Cada vez que cometo un error utilizando el sitio web, lo resuelvo fácil y rápidamente	☐	☐	☐	☐	☐	☐	☐
9 La información (como ayuda en línea, mensajes en pantalla y otra documentación) que provee este sitio web es clara.	☐	☐	☐	☐	☐	☐	☐
10 Es fácil encontrar en el sitio web la información que necesito.	☐	☐	☐	☐	☐	☐	☐
11 La información que proporciona el sitio web fue efectiva ayudándome a completar las tareas.	☐	☐	☐	☐	☐	☐	☐
12 La organización de la información del sitio web en la pantalla fue clara.	☐	☐	☐	☐	☐	☐	☐
13 La interfaz del sitio web fue placentera.	☐	☐	☐	☐	☐	☐	☐
14 Me gustó utilizar el sitio web.	☐	☐	☐	☐	☐	☐	☐
15 El sitio web tuvo todas las herramientas que esperaba que tuviera.	☐	☐	☐	☐	☐	☐	☐
16 En general, estuve satisfecho con el sitio web.	☐	☐	☐	☐	☐	☐	☐

Figura 10. Cuestionario CSUQ versión 3 en español.

En la Tabla 4 se muestra el promedio obtenido por cada una de las tres dimensiones.

Tabla 4. Promedios obtenidos por cada una de las dimensiones del cuestionario CSUQ.

Dimensión	Preguntas	Promedio
Calidad del sistema	1 a la 6	6.6
Calidad de la información	7 a la 12	5.97
Calidad de la interfaz y diseño	13 a la 15	6.53

Fuente: Elaboración propia.

Derivado de los puntajes obtenidos se presentan los siguientes hallazgos:

- La aplicación ofrece un buen índice de satisfacción del usuario, esto basándose en la calificación promedio de la aplicación de 6.40 puntos de 7 posibles.
- De acuerdo con los promedios, la dimensión peor calificada fue la calidad de la información, sin embargo, al realizar un cruce con las respuestas individuales se identifica que se debe a la falta de mensajes de error e información de ayuda, encontrándose un punto de mejora en este rubro.
- En esa misma dimensión, con base en los promedios individuales, se identifica que la información presentada para realizar las acciones primordiales para las que fue diseñada la aplicación se muestra correctamente.

Como se puede observar el aspecto de diseño está calificado de forma satisfactoria obteniendo 6.53 puntos de 7 totales, lo cual destaca la importancia de aplicar el DCU desde las primeras fases de desarrollo. Aun así, debido a que nos encontramos ante un proceso iterativo, al final de la prueba se solicitó a los usuarios realizar comentarios generales identificando los siguientes puntos de mejora:

- Colocar títulos en los ejes de los gráficos,
- Elementos informativos en los gráficos para que los usuarios novatos cuenten con ayuda,
- Un mejor posicionamiento de las flechas de retorno, y
- Un *spinner* de carga al momento de desplegar el código de los ejercicios.

Finalmente, la pregunta número 16 que hace a referencia a la satisfacción general del uso del sistema fue calificada con la totalidad de los puntos, es decir 7.

5. Conclusiones

Desde las primeras iteraciones del usuario con los *mockups* y prototipos de baja y alta fidelidad, se hayan muestras de satisfacción por parte de los profesores encuestados, sugiriendo que, a pesar de ser un desarrollo para modalidades no presenciales, este podría ser de utilidad incluso en modalidad presencial.

Acorde con los resultados expuestos en la sección anterior, se puede concluir que una herramienta de este tipo es de gran utilidad para los profesores dedicados a la enseñanza de la programación en modalidad mixta o a distancia, pues:

- Permite tener un panorama general del aprovechamiento del curso.
- Facilita la identificación de los alumnos con rezago.
- Las evidencias del trabajo de los alumnos se encuentran en la plataforma.

Esto se traduce en una mejor planificación de actividades e intervenciones para apoyar a los alumnos en riesgo; además, se logra un ahorro considerable de tiempo al no tener que descargar el código de cada uno de los alumnos para poder evaluarlo.

En relación con la continuidad del presente trabajo (trabajo futuro) existen diferentes alternativas para ampliar su alcance. En términos de Interacción Humano Computadora se plantea la posibilidad de normar las interacciones de los alumnos con la plataforma; en términos de educación se puede evaluar los resultados obtenidos derivados del uso; otra posibilidad es evaluar en extenso la UX tanto en alumnos como en docentes; y finalmente con base en la naturaleza iterante tanto de *learning analytics* como del DCU, se pueden buscar una mejora continua a lo ya presentado.

Como acotación, se puede notar que el DCU es una metodología que potencia el desarrollo de los productos de *software* y la satisfacción de los usuarios, pues como se reporta, no hubo necesidad de hacer grandes cambios entre el prototipo de alta fidelidad y el prototipo funcional, ahorrando tiempo y esfuerzo en etapas avanzadas de desarrollo.

6. Referencias

- [1] Fuentes-Rosado, J. I., Moo-Medina, M. (2017). Dificultades de aprender a programar. *Revista Educación en Ingeniería*, 12 (24), 76-82. <https://doi.org/10.26507/rei.v12n24.728>
- [2] Ghorashi, S., Jensen, C. (2017). Integrating collaborative and live coding for distance education. *Computer*, 50 (5), 27–35. <https://doi.org/10.1109/MC.2017.131>
- [3] Langton, J. T., Hickey, T. J., Alterman, R. (2004). Integrating tools and resources: A case study in building educational groupware for collaborative programming. *Journal of Computing Sciences in Colleges*, 19 (5), 140–153. <https://dl.acm.org/doi/10.5555/1060081.1060101>
- [4] Long, P., Siemens, G. (2011). *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*. Association for Computing Machinery. <https://dl.acm.org/doi/proceedings/10.1145/2090116>
- [5] Islam, M., Jin, S. (2019). *An overview of data visualization*. International Conference on Information Science and Communications Technologies (ICISCT). Tashkent, Uzbekistan. <https://doi.org/10.1109/ICISCT47635.2019.9012031>
- [6] Valero Sancho, J. L. (2014). La visualización de datos. *Ámbitos. Revista Internacional de Comunicación*, (25), 1-15. <http://ambitoscomunicacion.com/2014/la-visualizacion-de-datos/>
- [7] Few, S. (2010). *Dashboard design for at-a-glance monitoring*. Perceptual Edge.
- [8] Kitchenham, B. (2004). *Procedures for performing systematic reviews*. Keele University. <https://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf>
- [9] Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education*, 17 (3), 1–25. <https://doi.org/10.1145/3105910>
- [10] Ruiperez-Valiente, J. A., Gomez, M. J., Martínez, P. A., Kim, Y. J. (2021). Ideating and developing a visualization dashboard to support teachers using educational games in the classroom. *IEEE Access*, 9, 83467–83481. <https://doi.org/10.1109/ACCESS.2021.3086703>
- [11] Hundhausen, C. D., Olivares, D. M., Carter, A. S. (2017). Ide-based learning analytics for computing education. *ACM Transactions on Computing Education*, 17 (3), 1–26. <https://doi.org/10.1145/3105759>
- [12] Olivares, D., Hundhausen, C., Ray, N. (2022). Designing ide interventions to promote social interaction and improved programming outcomes in early computing courses. *ACM Transactions on Computing Education*, 22 (1), 1–29. <https://doi.org/10.1145/3453165>
- [13] Amo, D., Santiago, R. (2017). *Learning Analytics: La narración del aprendizaje a través de los datos*. Editorial UOC, S. L.
- [14] Jdoodle. (2024). *Execute code within your platform*. <https://www.jdoodle.com/integrate-online-ide-compiler-api-plugins>
- [15] Internacional Organization for Standardization. (2019). *Ergonomics of human-system interaction —Part 210: Human-centred design for interactive systems*. <https://www.iso.org/standard/77520.html>
- [16] Nielsen, J. (2000). *Why You Only Need to Test with 5 Users*. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- [17] Verbert, K., Govaerts, S., Duval, E., Santos, J. L., Assche, F. V., Parra, G., Klerkx, J. (2013). Learning dashboards: an overview and future research opportunities. *Personal and Ubiquitous Computing*, (18), 1499-1514. <https://doi.org/10.1007/s00779-013-0751-2>
- [18] Hedlefs Aguilar, M. I., de la Garza González, A., Sánchez Miranda, M. P., Garza Villegas, A. A. (2016). Adaptación al español del Cuestionario de Usabilidad de Sistemas Informáticos CSUQ. *RECI Revista Iberoamericana de las Ciencias Computacionales e Informática*, 4 (8), 84-99. <https://www.reci.org.mx/index.php/reci/article/view/35>