

Implementación del enfoque de cómputo físico con la tarjeta BBC Micro:Bit para la enseñanza de la programación

Juan Riquelme Odi

Universidad Autónoma de Querétaro, México

Claudia Marina Vicario Solórzano

Instituto Politécnico Nacional, México

Alexandro Escudero Nahón

Ma. Teresa García Ramírez

Universidad Autónoma de Querétaro, México

Resumen

Enseñar a programar es una tarea compleja debido a la necesidad de la comprensión de diferentes conceptos y elementos de las ciencias computacionales, así como de los lenguajes de programación. Existen diferentes enfoques y metodologías para enseñar de manera efectiva los fundamentos de la programación. Actualmente, uno de los enfoques más conocidos se llama cómputo físico, el cual combina elementos físicos (electrónicos) con componentes digitales (*software*). Este enfoque ha obtenido resultados favorables en la enseñanza de la programación y en otras áreas relacionadas con las ciencias computacionales. Esta investigación muestra los resultados de implementar el cómputo físico con la tarjeta BBC Micro:Bit y de generar contenidos con el modelo *Analysis, Design, Development, Implementation and Evaluation* (ADDIE) de diseño instruccional, en la Universidad Autónoma de Querétaro, en dos pruebas de concepto enfocadas en la enseñanza de los fundamentos de programación a través de cómputo físico en un contexto universitario.

Palabras clave

BBC Micro:Bit, ciencias computacionales, cómputo físico, fundamentos de programación, modelo ADDIE.

Implementation of the physical computing approach with the BBC Micro:Bit card for teaching computer applications programming

Abstract

Teaching computer applications programming is a complex task, because of the need to understand different concepts and elements of computer sciences, and also a programming language. There are different, approaches and methodologies to effectively teach the fundamentals of computer applications programming, but currently there is an approach known as physical computing, which combines physical components (electronic) with software components. This approach has obtained favorable results in the teaching of computer applications programming, and in other areas related to computer sciences. This research shows the results of implementing the physical computing with the BBC Micro:Bit pocket-sized computer, and generating

Keywords

ADDIE model, BBC Micro:Bit, computer sciences, fundamentals of computer applications programming, physical computing.

Recibido: 28/02/2020

Aceptado: 27/11/2020

content with the Analysis, Design, Development, Implementation and Evaluation (ADDIE) model of instructional design, at the *Universidad Autónoma de Querétaro*, in two concept proofs. Both proofs focused on teaching of the fundamentals of computer applications programming through physical computing in a university context.

Introducción

El aprendizaje efectivo de la programación de aplicaciones informáticas es una tarea compleja por su propia naturaleza, por situaciones que van desde el acercamiento de los estudiantes con los lenguajes de programación hasta la creación de códigos completos de varias líneas que resultan difíciles de depurar, corregir y optimizar para el funcionamiento adecuado de las aplicaciones generadas. Estos problemas son comunes con estudiantes que cursan las asignaturas vinculadas con la programación de aplicaciones informáticas ya que, generalmente, se desarrollan métodos tradicionales de enseñanza-aprendizaje para la gestión de este tipo de conocimientos. Esto provoca que los estudiantes pierdan el entusiasmo rápidamente, lo que genera apatía respecto de la programación y el desarrollo de aplicaciones informáticas (Jenkins, 2002).

Debido a esta problemática específica, es decir, a causa de que se suelen aplicar métodos de enseñanza tradicionales al momento de enseñar los principios de programación han surgido metodologías, técnicas, enfoques, iniciativas y herramientas tecnológicas para auxiliar en el proceso de enseñanza de la programación y, con ello, los estudiantes logren un aprendizaje efectivo. Una estrategia utilizada actualmente para la enseñanza de la programación es el uso del enfoque conocido como cómputo físico, el cual utiliza una combinación de medios físicos que son reales y tangibles para la enseñanza de las ciencias computacionales, que incluye la programación de aplicaciones informáticas, donde se busca una interacción física entre el individuo y el equipo de cómputo (Igoe, 2004).

Adicional a la estrategia requerida para la enseñanza de la programación, se necesita considerar la forma en la cual se conduce pedagógicamente el proceso de instrucción con los estudiantes, además de guiar el proceso de creación de contenidos para el mismo proceso de instrucción. Esta creación de contenidos requiere un modelo específico para asegurar la calidad y la efectividad pedagógica de dichos contenidos y, para ello, el modelo *Analysis, Design, Development, Implementation and Evaluation* (ADDIE) de diseño instruccional es uno de los más adecuados, ya que cada fase del modelo establece pertinentes acciones a considerarse para la generación de los contenidos requeridos. Este modelo se utiliza en la Universidad Autónoma de Querétaro para la creación de contenidos digitales de la enseñanza en línea, así como en los procesos de instrucción mixtos o combinados.

El objetivo de este artículo es presentar los resultados obtenidos mediante un diseño cuasi-experimental con dos grupos experimentales en cada cual se implemento uno de dos procesos de intervención como alteración de la variable independiente (dos pruebas de concepto), con el cual se persigue observar el aprendizaje de los fundamentos de programación por estudiantes, a través de la implementación del enfoque tecno-pedagógico de cómputo físico y mediante la utilización de la tarjeta BBC Micro:Bit como herramienta tecnológica en la Universidad Autónoma de Querétaro.

Marco teórico

La dificultad en el aprendizaje de la programación

El aprendizaje de la programación es una tarea considerada compleja, desde su concepción hasta su puesta en práctica, ya que la programación implica capacidad de abstracción para generar un producto intelectual, lógico e intangible que concluye con una aplicación informática. Una situación que refleja la dificultad de aprendizaje de la programación de aplicaciones informáticas es cuando los estudiantes logran dominar la sintaxis y la semántica de un lenguaje de programación específico y no logran el conocimiento necesario para trasladar todo ello al logro de una aplicación que funcione de manera correcta (Ala-Mutka, 2003).

Considerar que una persona es eficaz en la programación de aplicaciones informáticas significa que requiere un conjunto mínimo de habilidades específicas como el dominio de un lenguaje de programación, la capacidad de crear nuevos programas, depurar efectivamente el programa generado, reutilizar e integrar contenidos previos y otra serie de habilidades específicas de la programación (Isong, 2014).

Debido a este tipo de situaciones y habilidades requeridas, las personas estudiantes que inician sus estudios en la programación de aplicaciones informáticas pueden perder el entusiasmo rápidamente y abandonar el estudio de la programación, además de adquirir aversión hacia la programación de tales aplicaciones. Acorde con lo mencionado por Jenkins (2002), durante el proceso de enseñanza de la programación existe también la dificultad de que el estudiante desarrolle ciertas habilidades requeridas como la de crear un archivo con el código fuente del programa, de compilarlo o interpretarlo, de ejecutar la aplicación y llevar a cabo pruebas específicas para corregir los errores que pudieran surgir durante la ejecución, pues son éstas las habilidades que integran el marco holístico requerido para el aprendizaje de la programación de aplicaciones informáticas. Existen también

factores considerados problemáticos en relación con el aprendizaje de la programación de aplicaciones, uno de ellos es la falta de interés de algunos estudiantes atribuida a que en la mayoría de las asignaturas se les enseña más la sintaxis de un lenguaje de programación específico en lugar de desarrollar la lógica de la programación y enfatizar el cambio necesario en la forma de pensar.

En cuanto a la problemática sobre el aprendizaje de la programación de aplicaciones informáticas y el aprendizaje efectivo de los fundamentos de programación, ello ha sido estudiado por diferentes investigadores como Butler y Morgan (2007), Lahtinen, Ala-Mutka y Järvinen (2005), y Vera (2017). En todas estas investigaciones se especifican factores clave que determinan la no asimilación correcta de los conocimientos necesarios para la programación de aplicaciones. El primer factor considerado es el aprendizaje de conceptos abstractos específicos de la programación, como son la comprensión de los arreglos, las funciones, las matrices, etcétera. Además, la falta de contextualización del problema por resolver dificulta la asimilación de este tipo de conocimientos, pues la omisión de un problema real o práctico para aplicar los conocimientos adquiridos impide que los mismos puedan ser desarrollados plenamente y, asimismo, dificulta la aplicación real y práctica de los conocimientos y habilidades adquiridas durante todo el proceso de desarrollo de una aplicación informática.

Objetivo e hipótesis de la investigación

Debido a que se suelen aplicar métodos de enseñanza tradicionales al momento de enseñar los fundamentos de programación, han surgido diversos mecanismos para tratar de auxiliar en esta tarea compleja. Uno de los enfoques novedosos que se utiliza actualmente es el conocido como cómputo físico, el cual consiste en el uso de entornos físicos para que las personas puedan interactuar mejor con las computadoras (Igoe, 2004).

En este sentido, el objetivo de esta investigación fue aplicar dos procesos de intervención (dos pruebas de concepto) con la finalidad de observar el aprendizaje de los fundamentos de la programación con estudiantes a través de la implementación del enfoque tecno-pedagógico del cómputo físico y mediante la utilización de la tarjeta BBC Micro:Bit como herramienta tecnológica, en la Universidad Autónoma de Querétaro.

Por lo tanto, la hipótesis es: la aplicación del cómputo físico mediante la tarjeta BBC Micro:Bit por medio de una estrategia didáctica no tradicional facilita el aprendizaje de los fundamentos de programación en la Universidad Autónoma de Querétaro.

El cómputo físico como enfoque educativo

El cómputo físico incluye la construcción de sistemas físicos interactivos donde se busca conseguir una sinergia entre *hardware* y *software* para poder responder al mundo real y no solamente como lo sería con la forma de aprender e interactuar con la programación de aplicaciones informáticas desde una perspectiva tradicional. Acorde con la información de Tyn Can Learning (s. f.), el cómputo físico es un marco para la comprensión de las relaciones entre las personas y el mundo físico para lograr una interacción de los elementos electrónicos (físicos) y los sistemas de *software* (lógicos) para su vinculación. La finalidad del enfoque es convertir ideas abstractas en medios tangibles, productos de un proceso de pensamiento.

El enfoque de cómputo físico en contextos educativos puede llegar a presentar una vinculación con el construccionismo debido a que el aprendizaje es mucho más efectivo si el estudiante tiene la posibilidad de construir su propio conocimiento y desarrollar capacidades por su propia iniciativa, y cuando el aprendizaje adquiere relevancia personal para la persona que lo construye (Przybylla y Romeike, 2014a). Otra ventaja es que se obtienen productos tangibles en entornos físicos y reales, lo que permite el desarrollo de la creatividad e innovación por los estudiantes.

Al retomar al construccionismo junto con el cómputo físico, Blikstein (2015) especifica que se obtiene un aprendizaje en un contexto completo y robusto donde el estudiante se compromete con la construcción de un objeto tangible, público y que puede ser compartido y evaluado por sus compañeros, lo que permite construir el conocimiento y lograr un aprendizaje significativo y práctico.

Actualmente, existen varios dispositivos de cómputo físico utilizados en diferentes contextos de carácter educativo y académico, así como en el ámbito profesional e, inclusive, como pasatiempo. El abaratamiento de los dispositivos, la disponibilidad de componentes electrónicos y la facilidad de la programación de estos dispositivos han permitido que los mismos tengan un gran auge. Algunos dispositivos de cómputo físico utilizados hoy en día son la Tarjeta Programable Arduino, la Mini-Computadora Raspberry Pi y, recientemente, ha surgido una nueva tarjeta conocida con el nombre de BBC Micro:Bit, con gran éxito en Reino Unido y otros países.

Como tal, existen varias investigaciones que realzan el éxito del uso de cómputo físico como enfoque educativo para la enseñanza de las ciencias computacionales y la programación de aplicaciones informáticas, como son los estudios de Przybylla y Romeike (2014b), Rubio, Mañoso y Pérez (2013), y Schuls y Pinkwart (2015). En estas investigaciones se utilizaron herramientas de cóm-

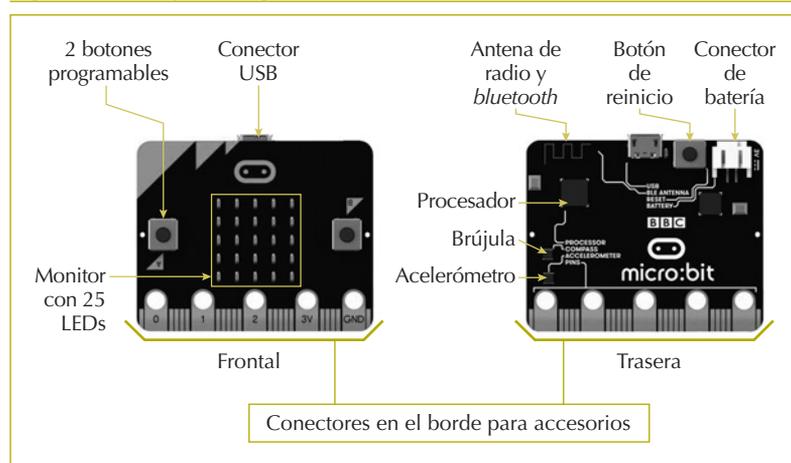
puto físico (como la tarjeta Arduino y LEGO Mindstorms) en las que los resultados arrojaron percepciones positivas de estudiantes y profesores con respecto al aprendizaje de las ciencias computacionales, la adquisición de un buen dominio de lenguajes de programación, el desarrollo de la motivación e, inclusive, se ha propuesto incorporar el cómputo físico en los currícula.

La tarjeta BBC Micro:Bit

La tarjeta programable BBC Micro:Bit fue creada en 2015 en el Reino Unido para fomentar la creatividad de niñas/os y adolescentes, y el desarrollo de habilidades de ciencias computacionales, tecnología e ingeniería (Sentance, Waite, Hodges, MacLeod y Yeomans, 2017). Acorde con la información de Ball et al. (2016), el dispositivo fue presentado en 2015 y luego se procedió a capacitar a profesores para que empezaran a utilizarlo e incorporarlo en el currículo escolar, y en 2016 se distribuyeron 800 000 dispositivos en Reino Unido, Wales, Irlanda del Norte y Escocia. Cabe destacar que con este dispositivo se busca que los estudiantes no sean solamente consumidores digitales, sino que tengan la posibilidad de ser creadores digitales y, con ello, aprovechen mejor sus capacidades (Roger et al., 2017). La tarjeta BBC Micro:Bit, con sus respectivos componentes electrónicos, se muestra en la **figura 1**.

La tarjeta cuenta con diferentes componentes para su utilización, en este caso un monitor frontal de 25 *Light Emitting Diode* (LED), dos botones, diecinueve terminales de entrada-salida, acelerómetro, compás, radio, sensor de luz, sensor de temperatu-

Figura 1. La tarjeta programable BBC Micro:Bit



Fuente: <https://microbit.org/es/guide/features/>

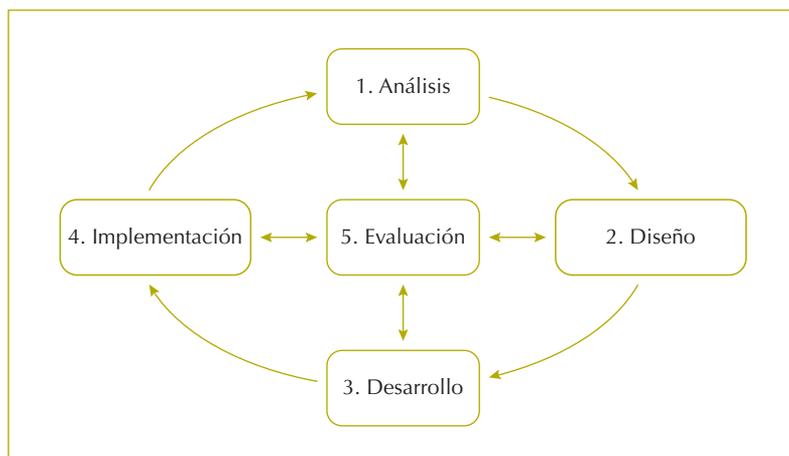
ra, vinculación vía *bluetooth* y una interfaz *Universal Serial Bus* (USB) para su programación, así como para el envío y recepción de datos entre el dispositivo y un equipo de cómputo (Monk, 2018). La programación del dispositivo se hace mediante un entorno de desarrollo disponible en la Internet llamado MakeCode Editor, el cual permite la programación del dispositivo mediante bloques, o bien, mediante código en JavaScript. Además, existe una versión de Python para la programación del dispositivo si se requiere utilizar características más avanzadas sobre el lenguaje de programación o por si ya se tiene conocimiento de este lenguaje.

La tarjeta BBC Micro:Bit ha sido objeto de estudio y, en varias investigaciones, se han registrado percepciones positivas, como se expresa en *British Broadcasting Corporation* (BBC, 2017), Carlborg y Tyrén (2017), Gibson y Bradley (2017), y Sentance et al. (2017). En estas investigaciones se han obtenido resultados positivos en relación con la utilización del dispositivo, tanto por parte de los profesores como de los estudiantes con respecto a la enseñanza de las ciencias computacionales, el desarrollo del pensamiento computacional, así como el aprendizaje de la programación de aplicaciones, inclusive mediante la realización de propuestas de modificar los programas de estudio de diferentes asignaturas para la incorporación didáctica de la tarjeta. En los resultados se obtuvo que el dispositivo es fácil de utilizar, promueve la imaginación y la creatividad, además de considerarse su utilización como una manera divertida de aprender en las asignaturas de carácter computacional.

Es necesario indicar que además se requiere un mecanismo que permita guiar el proceso de instrucción, así como la creación de los contenidos a utilizar para la enseñanza efectiva de los fundamentos de programación, y precisamente el modelo *Analysis, Design, Development, Implementation and Evaluation* (ADDIE) es uno de los más recomendables.

El modelo ADDIE para diseño instruccional

Para poder llevar a cabo un proceso de instrucción se requiere utilizar un modelo de diseño instruccional, con el objetivo de generar las actividades para alcanzar una acción de aprendizaje específica (Tobón, 2007), y el modelo ADDIE es uno de los más utilizados. Este modelo (también conocido como el modelo clásico) sigue una secuencia de actividades que permiten definir las necesidades en el proceso de instrucción, el diseño y la construcción de estas actividades a fin de obtener un producto final que cubra todas las características requeridas para la instrucción (Arshavskiy, 2014). Las fases esenciales de este modelo son las que se muestran en la **figura 2**.

Figura 2. Fases del modelo ADDIE

Fuente: Arshavskiy (2014, p. 13)

En este modelo se sigue un esquema lineal-secuencial, pero en todo momento se realizan acciones de evaluación para corregir aquellas actividades que no cumplen con los objetivos específicos por alcanzar, además de seguir un mecanismo iterativo-incremental para lograr la mejora del producto esperado. Una explicación más detallada de las fases de este modelo es la que se expone a continuación:

1. *Análisis*: identificación de los objetivos a alcanzar, los contenidos a ser incluidos y los resultados esperados de los materiales digitales.
2. *Diseño*: especificación de la forma de implementar la acción de formación y los materiales digitales.
3. *Desarrollo*: generación de los materiales digitales a utilizarse durante la instrucción.
4. *Implementación*: el contenido generado es dado a los estudiantes para su utilización y estudio.
5. *Evaluación*: valoración de la efectividad de la acción de formación, el cumplimiento de objetivos y el grado de aprendizaje de los estudiantes.

El modelo ADDIE se ha utilizado de forma exitosa en procesos de enseñanza en línea y combinados, y existen investigaciones como la de Estrada, Mendoza, Peraza, Tostado y Quirino (2013), Loor (2014), y la de Nauta, Platenkamp y Hettinga (2016) que han obtenido resultados exitosos con la implementación de contenidos digitales en cursos en línea y presenciales, mediante el uso del modelo ADDIE para crearlos, y en los resultados se identificaron también percepciones positivas de estudiantes y profesores,

así como una correcta asimilación de contenidos en los procesos de instrucción.

Método

El objetivo principal de esta investigación fue implementar dos procesos de intervención (dos pruebas de concepto) con la finalidad de observar el aprendizaje de los fundamentos de programación con estudiantes a través de la implementación del enfoque tecno-pedagógico del cómputo físico mediante la utilización de la tarjeta BBC Micro:Bit en la Universidad Autónoma de Querétaro.

Para lograr el objetivo de investigación, fueron implementados dos seminarios como concreción de sendas pruebas de concepto. La primera prueba de concepto estuvo enfocada en consolidar conocimientos de los fundamentos de programación con estudiantes que, por su trayectoria académica, podrían disponer de conocimientos previos, mientras que la segunda prueba fue encaminada con estudiantes que podrían no tener conocimientos previos. Los elementos característicos de ambas pruebas de concepto se muestran a continuación:

- ▶ *Prueba de concepto con estudiantes de los cuales se esperaba conocimientos previos*
 - *Objetivo:* observar la percepción y el aprendizaje de estudiantes sobre fundamentos de programación mediante el uso de cómputo físico, utilizando la tarjeta BBC Micro:Bit, en un seminario diseñado mediante el modelo ADDIE.
 - *Contexto de aplicación:* Facultad de Ingeniería de la Universidad Autónoma de Querétaro con estudiantes de 3.^{er} semestre de la carrera de ingeniería en automatización.
 - *Duración total del seminario:* 9 horas, divididas en 3 sesiones de 3 horas.
 - *Características de los estudiantes:* 17 en total, 14 hombres y 3 mujeres. 14 estudiantes manifestaron contar con conocimientos de programación y 3 afirmaron no contar con tales conocimientos.
- ▶ *Prueba de concepto con estudiantes de los cuales no se esperaba conocimientos previos*
 - *Objetivo:* observar la percepción y el aprendizaje de estudiantes sobre fundamentos de programación mediante el uso de cómputo físico, utilizando la tarjeta BBC Micro:Bit, en un seminario diseñado mediante el modelo ADDIE.
 - *Contexto de aplicación:* Facultad de Ingeniería de la Universidad Autónoma de Querétaro con estudiantes de semestre cero y de la licenciatura en matemáticas aplicadas.
 - *Duración total del seminario:* 12 horas, divididas en 3 sesiones de 4 horas.

- *Características de los estudiantes:* 13 en total, 5 hombres y 8 mujeres. 3 estudiantes expresaron contar con conocimientos de programación y 10 afirmaron no contar con tales conocimientos. 5 estudiantes pertenecían al semestre cero y 8 pertenecían a la licenciatura en matemáticas aplicadas.

Las pruebas de concepto implicaron la puesta en práctica de sendos seminarios (los cuales fueron prácticamente el mismo con excepción de las horas totales de duración), que dispusieron de contenidos diversos relacionados con los fundamentos de programación (Aguilar, 2008; Norton, 2006; Pressman, 2010; Zapata, 2013), otros relacionados con el cómputo físico (Platt, 2009; O'Sullivan e Igoe, 2004) y contenidos relacionados con el uso de la tarjeta BBC Micro:Bit (Halfacree, 2018; Monk, 2018). En el **cuadro 1** se muestran las unidades curriculares de los seminarios.

Los seminarios fueron presenciales, pero se aprovecharon varias herramientas digitales para que quienes asistieron pudieran acceder de manera asíncrona a contenidos digitales, por si acaso deseaban consultar a detalle alguno de estos. Los contenidos digitales de los seminarios fueron desarrollados a través del uso del modelo de diseño instruccional ADDIE, específicamente con las guías de Arshavskiy (2014) adaptadas al contexto de la Universidad Autónoma de Querétaro. Los contenidos digitales fueron ubicados en una plataforma en línea de gestión de aprendizaje.

Los contenidos incluyeron secciones teóricas respecto de los fundamentos de programación, otras enfocadas a componentes electrónicos, así como actividades propuestas para su ejecución, además se incluyeron secciones de comunicación en la plataforma en línea, por si era necesario resolver dudas que pudieran surgir a los estudiantes durante el acceso a los contenidos digitales.

Instrumentos de recolección de datos

La finalidad de los instrumentos y su funcionalidad de operativización es la siguiente:

- ▶ *Examen pre-test.* Utilizado para obtener información inicial de los estudiantes (sin relación con el objetivo de investigación que aquí se reporta), además de valorar sus conocimientos antes de cursar el seminario. Posteriormente, se ocupó esta información para obtener la diferencia antes y después de cursar el seminario. Este examen se construyó para valorar conocimiento de elementos teóricos de los fundamentos de programación, como lo son: estructuras secuenciales, estructuras de decisión, estructuras de iteración, variables y constantes, operadores, programación orientada a objetos y componentes electrónicos

Cuadro 1. Unidades curriculares de los seminarios

Número de unidad	Descripción
Unidad 1	Introducción a la tarjeta BBC Micro:Bit
Unidad 2	Estructuras de decisión
Unidad 3	Estructuras iterativas
Unidad 4	Sensores con la tarjeta BBC Micro:Bit
Unidad 5	Introducción a Python con la tarjeta BBC Micro:Bit
Unidad 6	Programación orientada a objetos e interconexiones
Unidad 7	Radiofrecuencia con la tarjeta BBC Micro:Bit
Unidad 8	Proyectos derivados con la tarjeta BBC Micro:Bit
Unidad 9	Proyecto final con la tarjeta BBC Micro:Bit

(Aguilar, 2008; Norton, 2006; Rubio et al., 2013; Zapata, 2013; véase el **anexo 1**).

- ▶ *Examen post-test*. Utilizado para valorar los conocimientos de los participantes después de cursar el seminario. Fue el mismo cuestionario que el examen *pre-test* (véase el **anexo 2**).
- ▶ *Cuestionario de escalamiento tipo Likert*. Fue aplicado para registrar la percepción de los estudiantes del seminario sobre los contenidos digitales en línea, las sesiones presenciales y demás componentes curriculares (Rubio et al., 2013). La escala del cuestionario fue desde el valor 1 (completamente en desacuerdo) hasta el valor 5 (completamente de acuerdo). Este cuestionario fue aplicado al finalizar cada seminario (véase el **anexo 3**).

Resultados y discusión

El examen *pre-test* y *post-test* fue el mismo. En la prueba de concepto con estudiantes de los cuales se esperaban conocimientos previos, se aplicaron el examen *pre-test* y *post-test* a los 17 estudiantes que cursaron el seminario, y se obtuvo una media de 63.65 en el *pre-test* y 74.81 en el *post-test* (medidas en la escala estandarizada de 0 a 100 puntos); con esto se registró un incremento nominal promedio de 11.16 puntos en los conocimientos de los estudiantes, medidos según el instrumento referido. Para determinar la significancia estadística de este incremento, se aplicó la prueba *t* de *student* para muestras relacionadas, con la obtención de un valor $p=0.001$.

Para la prueba de concepto con estudiantes de los cuales no se esperaban conocimientos previos, se aplicaron el examen *pre-test* y *post-test* a los 13 estudiantes que cursaron el seminario, con los cuales se obtuvo una media de 57.97 en el *pre-test* y 81.98 en

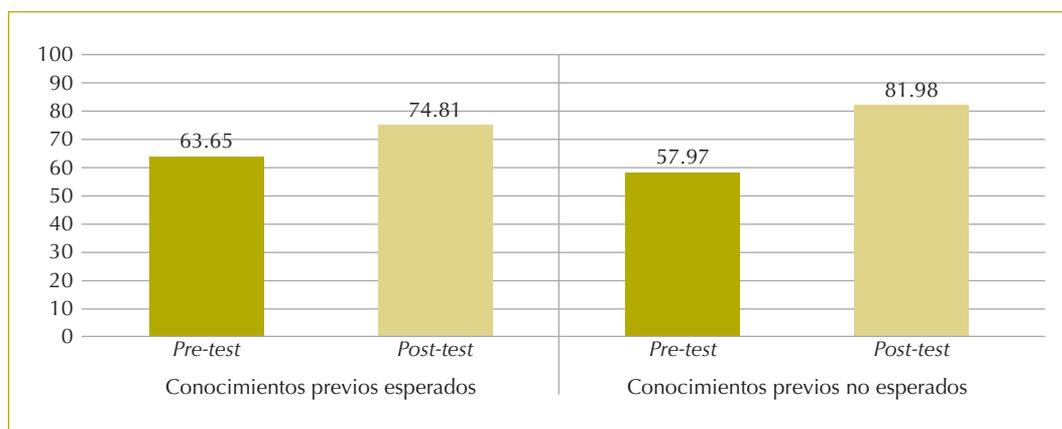
el *post-test* (igualmente medidas en la escala de 0 a 100 puntos), con lo que se se registró un incremento nominal promedio de 24.01 puntos en los conocimientos de los estudiantes, medidos según el instrumento referido. Al igual que en la otra prueba de concepto, se aplicó la prueba *t* de *student* y se obtuvo un valor $p < 0.001$, lo cual constituye evidencia empírica a favor de la hipótesis de incremento del conocimiento de los estudiantes sobre los fundamentos de programación asociado con el seminario cursado. En la **figura 3** se resumen las medias obtenidas para ambas pruebas de concepto, tanto en el examen *pre-test* como en el *post-test*.

Para ambas pruebas de concepto se determinó un incremento en los conocimientos de los fundamentos de programación y de los componentes electrónicos de los estudiantes, con esto se afirma que, para los casos concretos expresados anteriormente, la estrategia de cómputo físico a través de la implementación de los respectivos seminarios, permite a los estudiantes desarrollar los conocimientos de los fundamentos de programación, por tanto, se alcanzaron los objetivos planteados inicialmente en esta investigación.

Cabe destacar también que el grupo de estudiantes de los cuales no se esperaba conocimientos previos partió de una media menor en el examen *pre-test* y logró una mayor media en el *post-test*, por lo que un mayor número de horas curriculares puede asociarse con un mayor aprendizaje. También, es necesario señalar que la estrategia metodológica implementada en este estudio no comparó la efectividad de la estrategia pedagógica estudiada con otra cualitativamente distinta.

Por otro lado, los resultados obtenidos en esta investigación manifiestan cierto parecido con los de varios estudios previos, como los de Przybylla y Romeike (2014b), Rubio et al. (2013), y

Figura 3. Diferencias de medias entre *pre-test* y *post-test* por prueba de concepto



Schulz y Pinkwart (2015), con respecto a la ventaja de utilizar el cómputo físico como estrategia tecno-educativa para la enseñanza de la programación de aplicaciones informáticas y de las ciencias computacionales.

Las diferencias de medias entre *pre-test* y *post-test*, por prueba de concepto y dimensión de los instrumentos aplicados, se presentan en el **cuadro 2**.

En la prueba de concepto con estudiantes de los cuales se esperaba conocimientos previos, se presentó un incremento nominal promedio en seis de las ocho dimensiones del examen, pero sólo se presentó un incremento apreciablemente positivo en dos dimensiones, uno del 48.6% en la dimensión *E. Variables y constantes* y otro del 55.8% en la dimensión *F. Operadores*. Asimismo, puede interpretarse que no se presentó cambio alguno en el aprendizaje en relación con las dimensiones *A. Estructuras secuenciales* y *G. Programación orientada a objetos*, lo cual puede explicarse porque, a pesar de haberse indicado a los estudiantes que ingresarán a la plataforma digital y estudiarán los contenidos relacionados con la programación orientada a objetos, según los registros de la plataforma, sólo 8 de 17 estudiantes (47.1%) accedieron a estos.

Para la prueba de concepto con estudiantes de los cuales no se esperaba conocimientos previos, se presentó un incremento nominal promedio en siete de las ocho dimensiones del instrumento, de los cuales sólo en cuatro dimensiones fueron apreciablemente positivos, con un incremento de 53.6% en la dimensión *D. Tipos de datos*, otro de 48.5% en la dimensión *E. Variables y constantes*, uno de 108.0% en la dimensión *G. Programación orientada a objetos* y otro de 97.6% en la dimensión *H. Componentes electrónicos*. Asimismo, puede interpretarse que no se

Cuadro 2. Diferencias de medias para cada prueba de concepto por dimensión

Dimensión	Prueba de concepto			
	Conocimientos previos esperados		Conocimientos previos no esperados	
	<i>Pre-test</i>	<i>Post-test</i>	<i>Pre-test</i>	<i>Post-test</i>
A. Estructuras secuenciales	95.58	94.11	88.46	94.23
B. Estructuras de decisión	63.52	67.05	53.84	49.23
C. Estructuras de iteración	48.52	52.94	44.23	59.61
D. Tipos de datos	84.87	89.07	61.53	94.50
E. Variables y constantes	51.47	76.47	63.46	94.23
F. Operadores	50.58	78.82	75.38	84.61
G. Programación orientada a objetos	55.55	50.00	48.07	100.00
H. Componentes electrónicos	60.78	78.43	40.38	79.80

presentó cambio alguno en el aprendizaje en relación con la dimensión *B. Estructuras de decisión*, lo cual se atribuye, de igual manera, a la falta de acceso por los estudiantes a los contenidos específicos disponibles en la plataforma, donde sólo 5 de 13 estudiantes (38.5%) ingresaron a consultar estos contenidos.

Ambas pruebas de concepto observaron incrementos significativos en varias dimensiones relacionadas con los fundamentos de programación, por lo que se puede afirmar que el uso de cómputo físico es una estrategia adecuada para la enseñanza de este tipo de conocimientos que, por su naturaleza, pueden ser complejos y difíciles de enseñar. De esta forma, el uso de cómputo físico a través de un pertinente fundamento pedagógico y con la utilización de una herramienta tecnológica adecuada, permite comprender los fundamentos de programación.

El cuestionario de escala Likert fue aplicado al finalizar cada uno de los seminarios a un total de 17 estudiantes de los cuales se esperaba conocimientos previos y a 13 estudiantes de los cuales no se esperaba conocimientos previos. El cuestionario buscó analizar la percepción de los estudiantes respecto de los contenidos del seminario, las actividades propuestas, la utilización de la tarjeta BBC Micro:Bit como herramienta tecnológica, así como sobre el contenido digital y la funcionalidad de la plataforma en línea. Las medias obtenidas para cada prueba de concepto son las que se muestran en el **cuadro 3**.

Con los estudiantes de los cuales se esperaba conocimientos previos, se puede apreciar cómo en cuatro de las cinco dimensiones se presentaron medias superiores a los cuatro puntos, lo que permite expresar que tanto los contenidos de fundamentos de programación como los componentes electrónicos enseñados, el uso didáctico de la tarjeta BBC Micro:Bit y la misma percepción del seminario tuvieron promedios positivos, y el seminario en su conjunto fueron valorados positivamente por los estudiantes.

Con los estudiantes de los cuales no se esperaba conocimientos previos, se presentaron medias superiores a los cuatro puntos para las cinco dimensiones, por lo que, también en este caso los estudiantes valoraron positivamente el seminario tanto en su con-

Cuadro 3. Medias obtenidas a partir del cuestionario de escala Likert

Dimensión	Conocimientos previos esperados	Conocimientos previos no esperados
1. Fundamentos de programación	4.37	4.67
2. Componentes electrónicos	4.28	4.11
3. Tarjeta BBC Micro:Bit	4.36	4.94
4. Entornos de desarrollo	3.74	4.37
5. Percepción sobre el seminario	4.25	4.87

junto como cada uno de sus componentes, en especial los contenidos de fundamentos de programación y el uso didáctico de la tarjeta BBC Micro:Bit.

De manera concreta, la dimensión 5. *Percepción sobre el seminario* que, precisamente, se enfocó en la percepción sobre el proceso de instrucción, así como sobre los contenidos digitales de cada seminario, presentó las medias que se muestran en el **cuadro 4**.

Para la prueba de concepto con estudiantes de los cuales se esperaba conocimientos previos, en las subdimensiones 5.3. *Fundamentos de programación*, 5.4. *Claridad de materiales* y 5.6. *Acceso a contenidos* se presentaron medias cercanas a 4.5, por lo que, desde la perspectiva de los estudiantes, los contenidos y los materiales digitales fueron adecuados para el proceso de instrucción. Para la prueba de concepto con estudiantes de los cuales no se esperaba conocimientos previos, también se mostró una valoración positiva de los estudiantes con medias cercanas o iguales a 5 en todas las subdimensiones, lo cual expresa que los contenidos digitales y el proceso de instrucción fueron también valorados positivamente por los estudiantes.

A través de los resultados obtenidos se puede apreciar que el uso del modelo ADDIE en la Universidad Autónoma de Querétaro resultó ser un mecanismo pertinente para la generación de materiales digitales. Además, se guió eficazmente el proceso de instrucción, ya que las medias obtenidas a partir de los reactivos relacionados con el proceso de diseño instruccional resultaron altas.

Con el objetivo de considerar la confiabilidad de los instrumentos utilizados, se aplicaron diversas pruebas estadísticas. Específicamente, el *alfa* de Cronbach para el examen *pre-test* y *post-test*, así como para el cuestionario de escala Likert, y la prueba

Cuadro 4. Medias que se presentaron para la dimensión 5. *Percepción sobre el seminario*, del cuestionario de escala Likert

Subdimensión	Conocimientos previos esperados	Conocimientos previos no esperados
5.1. Métodos de enseñanza	3.94	4.92
5.2. Contenidos	4.05	4.85
5.3. Fundamentos de programación	4.41	5.00
5.4. Claridad de materiales	4.41	4.92
5.5. Lenguajes de programación	4.17	4.85
5.6. Acceso a contenidos	4.47	4.85
5.7. Sesiones presenciales	4.17	4.77
5.8. Ejercicios de unidad	4.35	4.85

KR-20 para el examen *pre-test* y *post-test*. Los resultados de estas pruebas se muestran a continuación en el **cuadro 5**. Entonces, los resultados derivados tanto del examen *pre-test* como del *post-test*, son preliminares por haber presentado una confiabilidad insuficiente. Mientras que los resultados del cuestionario de escala Likert son altamente confiables. Sin embargo, en general, todos estos resultados necesitan ser confirmados en futuras investigaciones que adicionen información sobre el grado de validez de estos instrumentos.

Conclusiones

Los resultados obtenidos en esta investigación muestran tentativamente que el cómputo físico, como enfoque tecno-pedagógico, fomenta el aprendizaje de los fundamentos de programación tanto con estudiantes con conocimientos previos como con aquellos sin estos conocimientos.

Para ambas pruebas de concepto se presentó un incremento estadísticamente significativo entre la media global del *pre-test* y del *post-test*, así como incrementos notables en algunas de las dimensiones del examen *pre-test* y *post-test*, como *E. Variables y constantes*, *F. Operadores* y *G. Programación orientada a objetos*.

La implementación del cómputo físico manifestó ser una forma concreta, quizás también lúdica, de enseñar los fundamentos de programación, donde los estudiantes aprenden mientras realizan proyectos físicos, lo cual favorece la integración con otros conocimientos. Por lo que se vislumbra la pertinencia de comparar su efectividad con la de la enseñanza tradicional, en futuras investigaciones.

Para cada prueba de concepto se presentaron resultados positivos en casi todas las dimensiones del cuestionario de escala Likert, a partir de lo cual, ahora desde la percepción de los estudiantes, se afirma que tanto cada seminario en su conjunto como los contenidos curriculares, el proceso de enseñanza, los materiales digitales y el uso de la tarjeta BBC Micro:Bit, en particular, fa-

Cuadro 5. Confiabilidad de los instrumentos

Conocimientos previos esperados		Conocimientos previos no esperados	
<i>Pre-test y post-test</i>		<i>Pre-test y post-test</i>	
Alfa de Cronbach	KR-20	Alfa de Cronbach	KR-20
0.7131	0.7315	0.7609	0.7812
Cuestionario de escala Likert		Cuestionario de escala Likert	
Alfa de Cronbach		Alfa de Cronbach	
0.9410		0.9142	

cilitan la enseñanza de los fundamentos de programación. Lo que permite finalmente concluir que la implementación del modelo ADDIE permitió desarrollar materiales digitales pertinentes.

Por último, se plantea continuar con la aplicación de este diseño de seminario en diferentes contextos educativos para establecer el grado de la validez de resultados que se obtuvieron en esta investigación. En este proceso pueden contemplarse actualizaciones pertinentes al material y a los contenidos del mismo, y con ello dar continuidad a la investigación sobre el cómputo físico como estrategia tecno-pedagógica para la enseñanza tanto de las ciencias computacionales como de los fundamentos de programación.

Agradecimientos

Se agradece al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo prestado para llevar a cabo esta investigación, así como por la adquisición de las tarjetas utilizadas en las implementaciones prácticas (con número de proyecto 294219). De igual forma, se agradece a la Red LaTE México y al Instituto Politécnico Nacional (con clave de proyecto SIP-2018-RE/048) por financiar también esta investigación en conjunto. Por último, se agradece al maestro Moisés Agustín Martínez Hernández y al maestro Iván González García, ambos de la Universidad Autónoma de Querétaro, por las facilidades prestadas para la implementación de ambos seminarios.

Se declara que la obra que se presenta es original, no está en proceso de evaluación en ninguna otra publicación, así también que no existe conflicto de intereses respecto a la presente publicación.

Referencias

- Aguilar, L. J. (2008). *Fundamentos de programación. Algoritmos, estructura de datos y objetos*. Madrid, España: McGraw-Hill.
- Ala-Mutka, K. M. (2003). *Problems in learning and teaching programming: a literature study for developing visualizations in the Codewitz-Minerva project*. Manuscrito inédito, Institute of Software Systems, Tampere University of Technology, Tampere, Finlandia.
- Arshavskiy, M. (2014). *Diseño instruccional para aprendizaje en línea. Guía esencial para la creación de cursos exitosos de educación en línea*. Charleston, EUA: Create-Space Independent Publishing Platform.
- Ball, T., Protzenko, J., Bishop, J., Moskal, M., Halleux de, J., Braun, M., Hodges, S., y Riley, C. (Mayo, 2016). *Microsoft touch develop and the BBC Micro:Bit*. Trabajo pre-

- sentado en la *38th International Conference on Software Engineering*, Austin, EUA. doi: 10.1145/2889160.2889179
- British Broadcasting Corporation (7 de julio de 2017). BBC Micro:Bit celebrates huge impact in first year, with 90% of students saying it helped show that anyone can code [Mensaje en un sitio digital en línea]. Recuperado de <https://www.bbc.co.uk/>
- Blikstein, P. (2015). Computationally enhanced toolkits for children: historical review and a framework for future design. *Foundations and Trends in Human-Computer Interaction*, 9(1), 1-68. doi: 10.1561/1100000057
- Butler, M., y Morgan, M. (Diciembre, 2007). *Learning challenges faced by novice programming students studying high level and low feedback concepts*. Trabajo presentado en la conferencia 2007 de la Australasian Society for Computers in Learning in Tertiary Education, Nanyang, Singapur.
- Carlborg, N., y Tyrén, M. (2017). *Introducing Micro:Bit in Swedish primary schools* (Tesis de maestría inédita). Chalmers University of Technology, Gotemburgo, Suecia.
- Estrada, R., Mendoza, R., Peraza, J. F., Tostado, M. I., y Quirino, L. G. (Noviembre, 2013). *Assessment of university students in a blended-learning model using ADDIE instructional design*. Trabajo presentado en la *6th International Conference of Education, Research and Innovation* organizada por la *International Academy of Technology, Education and Development*, Sevilla, España.
- Gibson, S., y Bradley, P. (2017). A study of Northern Ireland key stage 2 pupils' perceptions of using the BBC Micro:Bit in STEM education. *Student Teacher Perspectives*, 4(1), 15-41.
- Halfacree, G. (2018). *The oficial BBC Micro:Bit. User guide..* Indianapolis, EUA: John Wiley and Sons, Inc.
- Igoe, T. (4 de junio de 2004). What Is physical computing? [Mensaje en un diario digital]. Recuperado de <http://www.tigoe.com/blog/what-is-physical-computing/>
- Isong, B. (2014). A methodology for teaching computer programming: first year students' perspective. *International Journal of Modern Education and Computer Science*, 6(9), 15-21. doi: 10.5815/ijmecs.2014.09.03
- Jenkins, T. (Agosto, 2002). *On the difficulty of learning to program*. Trabajo presentado en la *3rd Annual LTSN-ICS Conference* organizada por la Loughborough University, Loughborough, Reino Unido.
- Lahtinen, E., Ala-Mutka, K. M., y Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *Association for Computing Machinery' Special Interest Group on Computer Science Education Bulletin*, 37(3), 14-18. doi: 10.1145/1151954.1067453
- Loor, K. (2014). Mejoramiento del trabajo autónomo de los estudiantes de la carrera licenciatura en Inglés de la Universidad Estatal Península de Santa Elena a través del curso Reading and Writing I con la modalidad *blended learning* 2012-2013. Experiencia pedagógica. *Revista Ciencias Pedagógicas e Innovación*, 1(3), 30-36.
- Monk, S. (2018). *Programming the BBC Micro:Bit. Getting started with MicroPython*. Nueva York, EUA: McGraw-Hill.
- Nauta, J. M., Platenkamp, J., y Hettinga, M. (Abril, 2016). *Development and evaluation of a blended learning course*. Trabajo presentado en la *Eighth International Conference on Mobile, Hybrid, and On-line Learning* organizada por la *International Academy, Research, and Industry Association*, Venecia, Italia.
- Norton, P. (2006). *Introducción a la computación*. Ciudad de México: McGraw-Hill.
- Platt, C. (2009). *Make: electronics. Learning through discovery*. California, EUA: Maker Media, Inc.
- Pressman, R. S. (2010). *Ingeniería del software, Un enfoque práctico*. Ciudad de México: McGraw-Hill.

- Przybylla, M., y Romeike, R. (2014a). *Key competences with physical computing*. Trabajo presentado en la *Key Competencies in Informatics, and Information and Communication Technology' Conference 2014* organizada por la Universität Potsdam, Potsdam, Alemania.
- Przybylla, M., y Romeike, R. (2014b). Physical computing and its scope. Towards a constructionist computer science curriculum with physical computing. *Informatics in Education*, 13(2), 225-240. doi: 10.15388/infedu.2014.14
- Rogers, Y., Shum, V., Marquardt, N., Lechelt, S., Johnson, R., Baker, H., y Davies, M. (2017). From the BBC Micro to Micro:Bit and beyond. *Interactions*, 24(2), 74-77. doi: 10.1145/3029601
- Rubio, M. A., Mañoso, C., y Pérez, Á. (Julio, 2013). *Using arduino to enhance computer programming courses in science and engineering*. Trabajo presentado en la *5th International Conference on Education and New Learning Technologies*, Barcelona, España.
- Schulz, S., y Pinkwart, N. (Noviembre, 2015). *Physical computing in STEM education*. Trabajo presentado en el *10th Workshop in Primary and Secondary Computing Education*, Londres, Reino Unido.
- Sentance, S., Waite, J., Hodges, S., MacLeod, E., y Yeomans, L. (Marzo, 2017). "Creating cool stuff": pupils' experience of the BBC Micro:Bit. Trabajo presentado en el Simposio Técnico 2017 organizado por el *Special Interest Group on Computer Science Education* de la *Association for Computing Machinery*, Seattle, EUA.
- O'Sullivan, D., y Igoe, T. (2004). *Physical computing: sensing and controlling the physical world with computers*. Boston, EUA: Thompson.
- Tobón, M. I. (2007). *Diseño instruccional en un entorno de aprendizaje abierto*. Manuscrito inédito, Facultad de Ciencias de la Educación, Universidad Tecnológica de Pereira, Pereira, Colombia.
- Tyn Can Learning (s.f.). Physical computing [Mensaje en sitio digital en línea]. Recuperado de <http://www.tyncan.com/physical-computing>
- Vera, E. E. (2017). *Estrategias de aprendizaje autónomo para disminuir los índices de reprobación en la materia de Metodología de la programación en la FCC de la BUAP en un sistema de e-learning* (Tesis doctoral inédita). Centro de Estudios Superiores en Educación, Ciudad de México, México.
- Zapata, C. A. (2006). *Fundamentos de programación, guía de autoenseñanza* (Tesis de licenciatura inédita). Universidad de Manizales, Manizales, Colombia.

Anexo 1. Fragmento del examen *pre-test* y *post-test*

Ejemplo de reactivos de la dimensión G. Programación orientada a objetos			
G.1. Se dice de aquellos elementos de una clase, la cual corresponde a las características de un objeto después de ser instanciado:			
a) Atributo	b) Método	c) Abstracción	d) Recursividad
G.2. Son los elementos de una clase, los cuales permiten ejecutar funcionalidad, haciendo que la clase pueda ejecutar operaciones:			
a) Atributo	b) Método	c) Abstracción	d) Herencia
G.3. Es el operador de una clase por el cual se puede acceder, ya sea a un atributo o a un método, directamente en un objeto:			
a) Más "+"	b) Menos "-"	c) Punto "."	d) Igual "="

Anexo 2. Fragmento del examen *pre-test* y *post-test*

Ejemplo de reactivos de la dimensión C. Estructura de iteración			
C.1. Un ciclo se repite de forma indefinida hasta que se cumple:			
a) Condición de paro	b) Condición de selección	c) Sentencial final	d) Incremento
C.2. Dado el siguiente algoritmo del factorial de un número, si "número=6" el resultado en "fact" será:			
Algoritmo factorial	Opciones:		
Definir numero, contador, fact como entero	a) 12		
Leer numero	b) 24		
fact=1	c) 120		
para contador=1 hasta numero	d) 720		
fact = fact * contador			
FinPara			
Escribir fact			
FinAlgoritmo			

Anexo 3. Fragmento del cuestionario de escala Likert

Ejemplo de reactivos de la dimensión 1. Fundamentos de programación				
1.2. El seminario me permitió comprender y utilizar correctamente las estructuras de decisión en programación:				
5. Definitivamente sí	4. Probablemente sí	3. Indeciso	2. Probablemente no	1. Definitivamente no
1.3. El seminario me permitió comprender y utilizar correctamente las estructuras iterativas en programación:				
5. Definitivamente sí	4. Probablemente sí	3. Indeciso	2. Probablemente no	1. Definitivamente no
1.4. El seminario me permitió comprender y utilizar correctamente las variables y constantes en programación:				
5. Definitivamente sí	4. Probablemente sí	3. Indeciso	2. Probablemente no	1. Definitivamente no
1.5. El seminario me permitió comprender y utilizar correctamente los tipos de datos en programación (enteros, reales, carácter, cadena, etc.):				
5. Definitivamente sí	4. Probablemente sí	3. Indeciso	2. Probablemente no	1. Definitivamente no