



EL PROCESO DE APRENDIZAJE E INNOVACIÓN EN EL SIGLO XXI; UNA EXPERIENCIA IBEROAMERICANA BASADA DESDE LA PERSPECTIVA DEL ALUMNO, PROFESOR Y TIC

COORDINADO POR:
Francisco Santillán Campos

Irene Aguilar Juárez
José Rubén Heredia Alonso
Paulina Salz Aguilar
Rosa Armida Zayas Bamerás
Jorge Arturo Castro Montoya
Martha Lorena Obermeier Pérez
José Manuel González Freire
Alma Isabel Arías Hurtado
Filberto Valenzuela Mendoza
Jesús Ramón López Sánchez
Raúl Bórquez Martínez
Sergio Alejandro Meza Olea
Jesús Hernández Del Real
Blanca Lila Ramos González
Leticia Del C. Romero Rodríguez
Gildardo Linarez Placencia
Flores González Samuel Jesús
Rosales Zapata Alicia Del Rocio
Luis Miguel Zapata Alvarado
Marco Antonio González Coronel
Lidia Meléndez Balbuena
José Albino Moreno Rodríguez
Alejandra Castro Lino
Guadalupe López Olivares
Roberto Ramírez Zavala
Sergio Magallanes
Brito Laredo Janette
Ferreira Martínez Vella Verónica
Garambullo Adriana Isabel
Eduardo César Contreras Delgado
Ana Luz Flores Pacheco
Jorge Horacio González Ortiz
David Gómez Sánchez
Ramón Gerardo Recio Reyes
Jesús René Guzmán Sánchez
Mariano Carillo Romero
María Eugenia Méndez
Héctor Hugo Zepeda Peña
Hugo Isaac Galván Álvarez
Rosario Lucero Cavazos Salazar

Tomas Day Sakis Fierro
Miguel Gastón Cedillo Campos
Giovanni Lidrongo Lidrongo
José Luis Soto Ortiz
Carlos Arturo Torres Gastelú
Aisida Leticia Tello Divicino
Fernando Gutiérrez Ortega
Santa Del Carmen Herrera Sánchez
Juan José Díaz Perera
Carlos Recio Urdaneta
Mario Saucedo Fernández
Rubén López Domínguez
Isabel López Zamora
Claudia Cinhya Peña Estrada
Luis Ambrosio Velázquez García
Erika Rivera Gutiérrez
Sandra Alicia Utrilla Cobos
Alejandro Higuera Zimbrón
Miguel Ángel Rubio Toledo
Ricardo Victoria Urbe
Claudia Araceli Figueroa Rochin
Jesús Adalberto Sepúlveda Rodríguez
Yessica Espinosa Díaz
América Fuentes Gálvez
Maribel Licona Labastida
Sandra López Reyes
Carolina Gómez Hinojosa
Rodolfo Mundo Velázquez
Blanca Hortencia Morales Vázquez
Martha Elva Restinza Ortega
Sara Morales García
Ima Araceli Mendoza Bello
Araceli Cuadras Urzuzástegui
Ima Araceli Bello Reyes
María Mima Flores Briseño
María Del Socorro Pérez Alcalá
Francisco Santillán Campos
Rosa María Michel Nava
Cynthia Alejandra Martínez Pinto
Venustiano Toledo Vázquez



El proceso de aprendizaje e innovación en el siglo XXI; Una experiencia Iberoamericana basada desde la perspectiva del alumno, profesor y TIC



www.umbra.com.mx



El proceso de aprendizaje e innovación en el siglo XXI; Una experiencia Iberoamericana basada desde la perspectiva del alumno, profesor y TIC

Francisco Santillán Campos - Coordinador

Irene Aguilar Juárez
José Rubén Heredia Alonso
Paulina Saiz Aguilar
Rosa Armida Zayas Barreras
Jorge Arturo Castro Montoya
Martha Lorena Obermeier Pérez
José Manuel González Freire
Alma Isabel Arias Hurtado
Filiberto Valenzuela Mendoza
Jesús Ramón López Sánchez
Raúl Bórquez Martínez
Sergio Alejandro Meza Olea
Jesús Hernández Del Real
Blanca Lilia Ramos González
Leticia Del C. Romero Rodríguez
Gildardo Linarez Placencia
Flores González Samuel Jesús
Rosales Zapata Alicia Del Rocío
Luis Miguel Zapata Alvarado
Marco Antonio González Coronel
Lidia Meléndez Balbuena
José Albino Moreno Rodríguez
Alejandra Castro Lino
Guadalupe López Olivares
Roberto Ramírez Zavala
Sergio Magallanes
Brito Laredo Janette
Ferreiro Martínez Velia Verónica
Garambullo Adriana Isabel
Eduardo César Contreras Delgado
Ana Luz Flores Pacheco
Jorge Horacio González Ortiz
David Gómez Sánchez
Ramón Gerardo Recio Reyes
Jesús René Guzmán Sánchez
Mariano Carrillo Romero
María Eugenia Méndez
Héctor Hugo Zepeda Peña
Hugo Isaac Galván Álvarez
Rosario Lucero Cavazos Salazar

Tomas Eloy Salais Fierro
Miguel Gastón Cedillo Campos
Giovanni Lizárraga Lizárraga
José Luis Soto Ortíz
Carlos Arturo Torres Gastelú
Aleida Leticia Tello Divicino
Fernando Gutiérrez Ortega
Santa Del Carmen Herrera Sánchez
Juan José Díaz Perera
Carlos Recio Urdaneta
Mario Saucedo Fernández
Rubén López Domínguez
Isabel López Zamora
Claudia Cintya Peña Estrada
Luis Ambrosio Velázquez García
Erika Rivera Gutiérrez,
Sandra Alicia Utrilla Cobos
Alejandro Higuera Zimbrón
Miguel Ángel Rubio Toledo
Ricardo Victoria Uribe
Claudia Araceli Figueroa Rochín
Jesúan Adalberto Sepúlveda Rodríguez
Yessica Espinosa Díaz.
América Fuentes Gálvez
Maribel Licona Labastida
Sandra López Reyes
Carolina Gómez Hinojosa
Rodulfo Mundo Velásquez
Blanca Hortencia Morales Vázquez
Martha Elva Reséndiz Ortega
Sara Morales García
Irma Araceli Mendoza Belío
Araceli Cuadras Urtuzuástegui
Irma Araceli Belío Reyes
María Mirna Flores Briseño
María Del Socorro Pérez Alcalá
Francisco Santillán Campos
Rosa María Michel Nava
Cynthia Alejandra Martínez Pinto
Venustiano Toledo Vázquez

Estimado profesor, si desea:

 Conocer otros materiales ✓

 Asesoría para elaborar un libro ✓

 Publicar un texto ✓

Acérquese a nosotros:

 Tel/fax: (0133) 31 33 30 53 y 31 33 30 59

 web: www.umbral.com.mx editorialumbral@hotmail.com

 Privada Porfirio Díaz N° 15 Col. El Mante
C.P. 45235 Zapopan, Jalisco, México

◆ Asesoría Editorial: Rafael Alejandro Orozco Díaz
alejandro.orozco@umbral.com.mx

◆ Comercialización: Ricardo Comparán Rizo
ricardo.comparan@umbral.com.mx

Desarrollo editorial y cubierta: Nohemí Aidé Guzmán Pérez

El proceso de aprendizaje e innovación en el siglo XXI; Una experiencia Iberoamericana basada desde la perspectiva del alumno, profesor y TIC

Derechos de autor:

© 2013, Irene Aguilar Juárez, José Rubén Heredia Alonso, Paulina Saiz Aguilar, et ál.

Coordinador: Francisco Santillán Campos,

© Umbral Editorial, S.A. de C. V.

ISBN: 978-607-8254-81-1

Miembro de la Cámara Nacional de la Industria Editorial Mexicana Socio # 3338

Umbral y su símbolo identificador son una marca comercial registrada.

Queda prohibida la reproducción o transmisión total o parcial del contenido de la presente obra mediante algún método, sea electrónico o mecánico (INCLUYENDO EL FOTOCOPIADO, la grabación o cualquier sistema de recuperación o almacenamiento de información), sin el consentimiento por escrito del editor.

Impreso en México / Printed in Mexico



MIGRACIÓN DE UNA PLATAFORMA DE APRENDIZAJE VIRTUAL: ESTRATEGIA, RETOS Y RECOMENDACIONES	19
Claudia Araceli Figueroa Rochín, Jesuán Sepúlveda Rodríguez, Yéssica Espinosa Díaz	
MODALIDADES EDUCATIVAS FRENTE AL RETO DE LA INNOVACIÓN EN EL APRENDIZAJE	27
Claudia Cintya Peña Estrada, Luis Ambrosio Velázquez García	
EMPLEO DE TIC EN EL PROCESO DE INVESTIGACIÓN EN EDUCACIÓN ESPECIAL	33
Eduardo César Contreras Delgado	
CENTRO DE INNOVACIÓN TECNOLÓGICA DE LA FACULTAD DE ARQUITECTURA Y DISEÑO, UAEMÉX	48
Érika Rivera Gutiérrez, Sandra Alicia Utrilla Cobos, Alejandro Higuera Zimbrón, Miguel Ángel Rubio Toledo, Ricardo Victoria Uribe	
EXPERIENCIAS EN LA IMPLEMENTACIÓN DE UN AULA VIRTUAL, EN DIFERENTES INSTITUCIONES DEL ESTADO DE ZACATECAS, MÉXICO	57
Alicia del Rocío Rosales Zapata, Luis Miguel Zapata Alvarado, Samuel Jesús Flores González	
INTERNET: NODO DE APRENDIZAJE EN LA SOCIEDAD DEL CONOCIMIENTO	63
Gildardo Linarez Placencia	
LA VINCULACIÓN Y EL PAPEL ESTRATÉGICO DE LOS ACTORES EN LA CONFORMACIÓN DE UN NUEVO MODELO EDUCATIVO DE LAS UNIVERSIDADES PARA EL SIGLO XXI	74
Jesús Hernández del Real, Blanca Lilia Ramos González, Leticia del C. Romero Rodríguez	
ESTRATEGIA DIDÁCTICA PARA EL MEJORAMIENTO DEL APRENDIZAJE APLICADO A LA MATERIA DE MANTENIMIENTO DE SISTEMAS HIDRÁULICOS Y NEUMÁTICOS	95
Jesús René Guzmán Sánchez, Mariano Carrillo Romero	
PERSPECTIVA DE LOS ACTORES SOBRE UNA UNIVERSIDAD VIRTUAL BASADA EN EL ENFOQUE DE COMPETENCIAS	107
José Luis Soto Ortiz, Carlos Arturo Torres Gastelú	



LA EVALUACIÓN EN TECNOLOGÍA EDUCATIVA	116
Héctor Hugo Zepeda Peña, Hugo Isaac Galván Álvarez, María Eugenia Méndez	
RECURSOS TECNOLÓGICOS PARA LA EDUCACIÓN Y USO DE PLATAFORMAS DE APOYO A LA ENSEÑANZA DE LENGUAS EN TRES NIVELES EDUCATIVOS.	122
Martha Lorena Obermeier Pérez	
RETOS Y PERSPECTIVAS DE FAADER, UAS, PARA TRABAJAR BAJO LA MODALIDAD B-LEARNING.	131
Paulina Saiz Aguilar, Rosa Armida Zayas Barreras, Jorge Arturo Castro Montoya	
MEJORA DEL APRENDIZAJE DE PROGRAMACIÓN ORIENTADA A OBJETOS A TRAVÉS DE UNA PLATAFORMA WEB.	139
Rosa María Michel Nava, Cynthia Alejandra Martínez Pinto	
EL ROL DEL PROFESOR TUTOR EN LOS PROGRAMAS DE EDUCACIÓN A DISTANCIA FIME-UANL.	150
Rosario Lucero Cavazos Salazar, Tomás Eloy Salais Fierro, Giovanni Lizárraga Lizárraga, Miguel Gastón Cedillo Campos	
HACIA UNA NUEVA CULTURA DE LA ENSEÑANZA-APRENDIZAJE EN CIENCIAS: LAS TIC, LOS CONTENIDOS Y LA MOTIVACIÓN COMO PILARES FUNDAMENTALES PARA UNA INNOVACIÓN EDUCATIVA. . . .	156
Rubén López Domínguez, Isabel López Zamora	

Parte 2

Aprendizaje e innovación en el aula

ESTRATEGIA TEÓRICO-METODOLÓGICA PARA FUNDAMENTAR EL USO DE LAS TIC EN EL EJERCICIO EDUCATIVO	173
Maribel Licona Labastida, América Fuentes Gálvez	
APOYAR EL APRENDIZAJE, UNA TAREA DE LA ORIENTACIÓN EDUCATIVA.	189
Ana Luz Flores Pacheco	
LA TRANSVERSALIDAD DE LOS MODELOS EDUCATIVOS COMO PUNTO NEURÁLGICO DE UN APRENDIZAJE SIGNIFICATIVO.	196
Blanca Hortencia Morales Vázquez, Martha Elva Reséndiz Ortega, Sara Morales García	



LA SIMULACIÓN EN EL APRENDIZAJE BASADO EN COMPETENCIAS DE LA PROGRAMACIÓN ESTRUCTURADA	205
Irene Aguilar Juárez, José Rubén Heredia Alonso	
ESTUDIO EXPLORATORIO SOBRE EL COMPORTAMIENTO CREATIVO EN DOCENTES DEL ÁREA DE LA SALUD DE LA UNIVERSIDAD AUTÓNOMA DE SINALOA	220
Irma Araceli Mendoza Belío, Araceli Cuadras Urtuzuástegui, Irma Araceli Belío Reyes	
ESTUDIO SOBRE LA ACTITUD HACIA LA NUEVA TECNOLOGÍA EDUCATIVA EN DOCENTES DEL NIVEL EDUCATIVO SUPERIOR	230
Jorge Horacio González Ortiz, David Gómez Sánchez, Ramón Gerardo Recio Reyes	
LA ENSEÑANZA DE LOS MEXICANISMOS COMO L2-LE EN EL SIGLO XXI.	240
José Manuel González Freire	
EL APRENDIZAJE DE CONCEPTOS BÁSICOS DE OPERACIONES UNITARIAS PARA ALUMNOS CON LA MATERIA DE TECNOLOGÍA FARMACÉUTICA USANDO LAS TIC	248
Marco A. González Coronel, Lidia Meléndez Balbuena, José Albino Moreno Rodríguez, Alejandra Castro Lino, Guadalupe López Olivares	
CONJUNTO DE VARIABLES COMO PROPUESTA PARA UNA EVALUACIÓN CUALITATIVA DEL APRENDIZAJE EN EL AULA A NIVEL SUPERIOR	255
Roberto Ramírez Zavala	
SOFTWARE EDUCATIVO EN SOPORTE A LA LECTURA Y ESCRITURA CHO'L- ESPAÑOL PARA ESCUELA PRIMARIA BILINGÜE	272
Sandra López Reyes, Carolina Gómez Hinojosa, Rodolfo Mundo Velásquez	
LOS RECURSOS DIDÁCTICOS EN EL DESARROLLO DE LA COMPETENCIA SOLUCIÓN PROBLEMA	281
Santa del Carmen Herrera Sánchez	
LA COMUNIDAD DE APRENDIZAJE: UNA EXPERIENCIA DE LA FORMACIÓN ACADÉMICA CON RECURSOS DE LAS NTIC	288
Venustiano Toledo Vázquez	

LA SIMULACIÓN EN EL APRENDIZAJE BASADO EN COMPETENCIAS DE LA PROGRAMACIÓN ESTRUCTURADA

Irene Aguilar Juárez, José Rubén Heredia Alonso

RESUMEN

Aprender a programar computadoras es una habilidad necesaria en todos los perfiles relacionados con la informática y la computación, es importante porque desarrolla habilidades como la abstracción, la capacidad de análisis, la especificación de problemas, el proponer, implementar y evaluar soluciones. La complejidad de desarrollar estas competencias profesionales sobre todo en personas que nunca han programado genera la necesidad de innovar métodos de enseñanza que incorporen nuevas herramientas de software como los simuladores. De igual forma es necesario implementar instrumentos de evaluación que faciliten a los docentes medir el aprendizaje logrado por el alumno. Las ventajas que ofrecen los simuladores en la comprensión de los sistemas y su funcionamiento son numerosas y pueden aprovecharse en el desarrollo de las habilidades y conocimientos de los estudiantes, especialmente los que requiere un alto nivel de abstracción. En este trabajo se presenta una propuesta didáctica para usar simuladores en el aula, con la finalidad de que el alumno adquiera habilidades y conocimientos de programación estructurada. La propuesta consta de dos elementos; las estrategias de enseñanza basada en competencias, integradas por actividades de aprendizaje centradas en el alumno y el diseño de los instrumentos de evaluación basados en rúbricas y una matriz de valoración de competencias

PALABRAS CLAVE: simulación, computación, competencias procedimentales, instrumentos de evaluación.

INTRODUCCIÓN

Los simuladores son programas especializados que permiten actuar virtualmente del mismo modo que en la realidad, facilitan la vinculación del conocimiento nuevo con la experiencia, además permiten cometer errores sin provocar daños en equipos costosos y tomar experiencia sobre el uso de equipo sofisticado a un precio asequible, estas ventajas hacen de los simuladores una herramienta muy útil para el aprendizaje de las ciencias experimentales y de las ingenierías, las cuales requieren una constante capacitación en el uso de equipo, costoso, escaso y delicado.

Para la formación profesional de un ingeniero en computación es necesario estudiar con un alto nivel de profundidad ocho áreas de conocimiento: entorno social, matemáticas, arquitectura de computadoras, redes, software de base, programación e ingeniería de software, tratamiento de la información, e interacción hombre-máquina. En el área de programación e ingeniería del software es necesario que el alumno aprenda a programar computadoras, esta habilidad es importante porque le permite proponer, implementar y evaluar soluciones en el ámbito laboral. En este trabajo se analizan distintos métodos de enseñanza de la programación, se evalúan las competencias necesarias para programar respecto a la taxonomía de Bloom, y se propone una metodología de enseñanza basada en el uso de simuladores para un curso de Programación Estructurada, dirigido a personas con solo conocimientos básicos de computación.

CONTENIDO

1.1. ANTECEDENTES

La licenciatura de Ingeniería en Computación pertenece a los perfiles profesionales incluidos en el dominio de la Informática y Computación, de acuerdo con el “Comité de Modelos Curriculares, Nivel licenciatura, Informática y Computación”, formado en el interior de la ANIEI (Asociación Nacional de Instituciones de Educación en Informática, A.C.), junto con la licenciatura en Informática, la licenciatura en Ingeniería del Software y la licenciatura en Ciencias Computacionales.

Estos cuatro perfiles profesionales comparten ocho áreas del conocimiento importantes para la formación de profesionistas sólidamente preparados, las cuales son: entorno social, matemáticas, arquitectura de computadoras, redes, software de base, programación e ingeniería de software, tratamiento de la información, e interacción hombre-máquina (ANIEI, 2012); la diferencia entre los perfiles se presenta en el nivel de profundidad con que cada perfil trabaja sobre las áreas de conocimiento.

Para las áreas de conocimiento: programación e ingeniería de software, tratamiento de la información e interacción hombre-máquina, es muy importante desarrollar la habilidad de programar computadoras, sin embargo, a pesar de la importancia de esta habilidad, las asignaturas de programación presentan altos índices de reprobación en todo el país. Por citar un ejemplo, en la Universidad Autónoma de Aguascalientes la materia de Programación I de la licenciatura de Ingeniería en Sistemas Computacionales tiene 63.64% de reprobación, en Programación II se presenta 76.47%, en Ingeniería de Software reprobó 62.07% de los alumnos (DEI-UAA, 2009).

Autores como Mora, Pérez y Rodríguez (2010: 4); Villalobos (2009) y Michel (2011) reconocen y coinciden sobre la dificultad que enfrentan los alumnos ante las asignaturas de Programación. En palabras de Villalobos (2009) los alumnos “deben aprender a entender un problema (abstraer, modelar, analizar), a plantear soluciones efectivas (reflexionar sobre una abstracción, definir estrategias, seguir un proceso, aplicar una metodología, descomponer en subproblemas), a manejar lenguajes para expresar una solución (codificar, entender y respetar una sintaxis), a utilizar herramientas que entiendan esos lenguajes (programar, compilar, ejecutar, depurar), a probar que la solución sea válida (entender el concepto de corrección y de prueba), a justificar las decisiones tomadas (medir, argumentar), etcétera.”

Las estrategias de enseñanza de programación que se han usado tradicionalmente no facilitan el progresivo dominio del aprendizaje en el alumno. Satorre, Lorens y Puchol (1996) observan que “Habitualmente, los problemas son explicados como algo que se sabe hacer, como algo cuya solución se conoce y que no genera dudas ni exige tentativas: el profesor conoce la situación –para él no es un problema– y la explica linealmente, con toda claridad; consecuentemente, los alumnos pueden aprender dicha solución y repetirla ante situaciones idénticas, pero no aprenden a abordar un verdadero problema y cualquier pequeño cambio les supone dificultades insuperables”. Con esta situación, la evaluación de las competencias del alumno resulta compleja para el alumno y para el docente, pues frecuentemente las actividades, tanto como la evaluación no están pensadas en la valoración de las evidencias de una progresiva adquisición de las competencias en el alumno. Además ante la dificultad de la programación el alumno cae en la generación de código por prueba y error sin una comprensión adecuada del problema, de los conceptos o del algoritmo.

Villalobos (2009) identifica que la enseñanza tradicional de la programación se basa en la hipótesis de que el estudiante es capaz de generar las habilidades necesarias tratando de imitar lo que el profesor hace en el pizarrón. Esto tiene muchos problemas implícitos: la atención del aprendizaje se centra más en la sintaxis de un lenguaje que en la habilidad de especificar, analizar y resolver problemas, pues al alumno se le limita a ver cómo se resuelve un problema en el pizarrón, muchas veces el docente no analiza el problema. El aprendizaje depende de la capacidad de imitación del alumno, no se trabaja sobre la generalización que hace el alumno por lo que se pueden generar patrones de conducta erróneos difíciles de corregir, se genera la idea de que la programación “no es enseñable”.

Las actividades propias de la programación requieren de conocimientos de tipo procedimental, resultado de procesos cognitivos de análisis, evaluación y creación; los conocimientos y procesos cognitivos más complejos de la taxonomía de Bloom revisada (Blanco, 2007). Para que un alumno logre desarrollar programas de forma eficiente, antes debe lograr actividades de complejidad menor. En opinión de Buck (2000, en Martínez, 2005), cuando no se ha pasado por cada uno de los niveles se vuelve complejo para el alumno simular el proceso (o algoritmo) en su mente y por lo tanto plasmarlo en un programa.

Nivel	Actividad que se espera de un alumno que esté en este nivel
1. Conocer	Saber cuáles son los estatutos existentes y su sintaxis
2. Comprender	Leer un conjunto de instrucciones y decir qué hacen paso a paso. Leer un programa y decir qué hace.
3. Aplicar	Usar elementos de una biblioteca de funciones. Poder aplicar un estatuto para generar un efecto deseado.
4. Analizar	Leer un programa con el objetivo de hacerle cambios. Puede ser para corregir errores o para realizar una modificación a su funcionamiento. Traducir un programa a un diagrama de flujo.
5. Sintetizar	Escribir un programa dados los requerimientos.
6. Evaluar	Comparar dos programas dados que resuelven el mismo problema.

Tabla 1: Actividades de programación relacionadas con la taxonomía de Bloom

Según Buck (2000, en Martínez, 2005), las habilidades que un alumno manifiesta de acuerdo con su progresivo dominio de la programación corresponden a niveles cada vez más altos en la taxonomía de Bloom.

Se han hecho variados esfuerzos por abordar la problemática, autores como Satorre y cols. (1996) proponen una herramienta que visualiza la traza de los diagramas de flujo llamada “Programin en el laberinto”, con esta herramienta el alumno introduce un conjunto de sentencias, las cuales se evalúan sintácticamente y si no se presentan errores, se genera el diagrama de flujo y se visualiza la ejecución del programa. La misma técnica es usada por Martínez (2005) quien introduce el uso de RAPTOR, herramientas de software visualizador de la traza de ejecución de diagramas de flujo, la herramienta es similar a Programin en el laberinto, la diferencia radica en la calidad de la animación y la distribución de la herramienta.

Michel (2011), explica el uso de “MCC Lumin” (Metodología Constructivista para el lenguaje Lumin), esta herramienta posee un lenguaje de programación orientado a objetos en español, un compilador que genera código objeto transportable y un entorno de desarrollo amigable. Esta metodología tiene la finalidad de facilitar el aprendizaje de la programación para cualquier persona sin conocimientos previos, de manera sencilla y agradable.

Mora y cols. (2010) reportan los resultados favorables en el aprendizaje de la programación mediante el uso de un juego que permite al alumno resolver un laberinto por medio de instrucciones, de esta forma el alumno aprende las estructuras sintácticas de un lenguaje de programación. En sus resultados se observa una buena aceptación de los alumnos, un incremento en calificaciones y la reducción del índice de reprobación, el cual baja de 49% a 11%.

Las soluciones a este problema también surgen en el ámbito internacional, Villalobos (2009), publica sus trabajos de investigación sobre esta problemática en la Universidad de los Andes y describe su estrategia de solución: el proyecto Cupi2, que se origina desde el año 2004 y que ha transitado por siete etapas:

1. Diagnóstico
2. Definición del marco conceptual, las hipótesis, las variables, los indicadores y los instrumentos
3. Diseño de la propuesta
4. Validación de las hipótesis por medio de pilotos controlados
5. Construcción de los materiales y herramientas de apoyo
6. Proceso de formación y acompañamiento de profesores
7. Creación de una comunidad de aprendizaje y difusión de resultados.

Este proyecto se ha convertido en eje de referencia para el cambio curricular de muchos programas de Ingeniería de Sistemas en Colombia, la propuesta se basa en el uso de software de representación y animación de algoritmos, unido a estrategias de aprendizaje basadas en el aprendizaje activo, aprendizaje incremental y aprendizaje basado en problemas.

Ante este contexto surge la inquietud de abordar en el centro Universitario UAEM Texcoco la enseñanza de la programación de una forma creativa, considerando actividades que promuevan los procesos cognitivos necesarios para el análisis de problemas, el planteamiento y evaluación de algoritmos, la complejidad progresiva de los procesos cognitivos que el alumno debe lograr, la adecuada evaluación formativa de las competencias en desarrollo, sin ignorar la autoevaluación que permita al alumno reconocer su avance, para conservar la motivación y perseverancia en su aprendizaje.

1.2. SIMULADORES

La simulación es una técnica numérica que mediante la modelación de los sistemas reales, permite imitar el comportamiento de las variables y sus interrelaciones, para comprender los procesos internos y modificarlos si es necesario.

Un simulador es una máquina que reproduce el comportamiento de un sistema en ciertas condiciones. Los simuladores suelen combinar partes mecánicas o electrónicas y partes virtuales que le ayudan a simular la realidad, por lo tanto, pueden utilizarse en el ámbito profesional o como un instrumento de ocio y entretenimiento. En la formación profesional los simuladores son indispensables para la formación de personas que tendrán una gran responsabilidad a su cargo, ya que sus eventuales errores pondrían en riesgo la vida de terceros o la funcionalidad de equipo altamente costoso. Gracias al simulador, los alumnos pueden entrenarse hasta adquirir la experiencia y las destrezas necesarias para desempeñarse profesionalmente. Si cometen errores en un simulador, nada saldrá dañado.

Una aplicación común de la simulación en la enseñanza es el laboratorio virtual, que es un sitio informático que simula una situación de aprendizaje comúnmente realizada en un espacio físico llamado laboratorio (Monge-Najera, Méndez-Estrada

y Rivas-Rossi 2005). De forma más específica, los laboratorios virtuales son simuladores que modelan un laboratorio real o un experimento, se ejecuta en una computadora, se consideran simuladores porque físicamente no existen los equipos, los dispositivos ni los materiales necesarios para ejecutar el experimento, sin embargo, la modelación del fenómeno y los componentes del simulador permiten repetir múltiples veces los experimentos, con diversos parámetros, hasta que se comprenden los principios de funcionamiento del sistema.

Los laboratorios virtuales pueden usarse como un complemento de los laboratorios convencionales, uno de los beneficios de los laboratorios virtuales es el control del tiempo, pues en la simulación los eventos pueden transcurrir con lapsos y frecuencias de tiempos diversas, sin consumir reactivos o materiales costosos.

El uso de los laboratorios virtuales data desde los años 50, cuando después de la Segunda Guerra Mundial se simuló sobre fenómenos de energía y física nuclear. Algunos de los simuladores se desarrollan con técnicas de realidad virtual, razón por la cual, estos simuladores adquieren las cuatro propiedades que de acuerdo con Pino-González citado por Mancio (2006) ofrecen los sistemas basados en realidad virtual, las cuales son:

1. **La capacidad sintética:** El mundo virtual se genera en tiempo real según sea alimentado por las instrucciones y/o acciones del usuario.
2. **la interactividad:** el entorno responde a las acciones y movimientos del usuario.
3. **La tridimensionalidad:** El entorno se desenvuelve en cualquier dispositivo de despliegue, imitando un mundo en tres dimensiones.
4. **La ilusión de realidad:** Los entornos que generan los sistemas de realidad virtual no son reales, sino más bien ficticios, pero son definidos, de tal modo que al usuario le parezcan reales.

Monge-Nájera y cols. (2005), señalan que técnicamente los laboratorios virtuales se basan exclusivamente en códigos de programación como HTML, Java; programas que son libres de pago de licencia, lo que permite un desarrollo económico para las instituciones, los programas se basan en el uso de hipertexto, lo que facilita su uso en computadoras básicas, con el único requisito de tener instalado un navegador web, el cual generalmente es fácil de adquirir o ya viene incluido en el sistema operativo.

Ventajas en la educación

El proceso de la simulación para el aprendizaje y el desarrollo de competencias tiene las siguientes ventajas según Ruiz (n, d):

- El simulador se convierte en un mecanismo dinámico de aprendizaje y fortalecimiento de competencias generales y específicas.
- Utiliza herramientas que fortalecen el aprendizaje de lo que es y será útil en situaciones reales.
- Ofrece al participante una experiencia práctica, que le permite decidir cómo y cuándo utilizar un procedimiento estratégico adecuado.
- Crea una motivación e interacción al aprovechar la simulación como herramienta que permite mejorar el aprendizaje individual.
- Brinda al participante la oportunidad de evaluar rápidamente los resultados de sus propias actuaciones y reorientar sus decisiones futuras.
- Permite la optimización del tiempo, y anular el riesgo de comprometer recursos reales en el proceso de toma de decisiones a través de la simulación.

La simulación otorga experiencia para los estudiantes, y los beneficios que ello implica. En palabras de Randolph Frederick Pausch, Ingeniero de Sistemas; uno de los líderes de la tecnología virtual y profesor de la universidad estadounidense de Carnegie Mellon, “Si no consigues tus sueños, puedes conseguir mucho intentando conseguirlos. Experiencia es lo que tú consigues, cuando no consigues lo que quieres”, Ruiz (n, d).

En opinión de López de Munain (2005), la Simulación permite entre otras cosas:

- El Aprendizaje por Descubrimiento:** es una forma activa de aprender en la que el alumno es el propio artífice de su aprendizaje. El método de aprendizaje ensayo-error es perfectamente compatible con esta dimensión de los sistemas de simulación.
- Fomentar la creatividad:** es una de las ventajas de los entornos de simulación. La posibilidad de disponer de “ToolBoxes & Blocksets” (cajas de herramientas y colecciones de bloques-operadores) en los entornos permite la disponibilidad de un

laboratorio, taller, o mesa de diseño con la que el alumno pueda no solo simular modelos que se le den hechos sino que pueda construir los suyos propios.

Ahorrar tiempo y dinero: procesar la información no es tarea fácil, y la adquisición, ordenación, tratamiento y análisis de la información son aspectos muy importantes de cara al proceso de aprendizaje. La simulación facilita la construcción de los modelos, y el tratamiento repetitivo de los datos.

La enseñanza individualizada: las herramientas de simulación permiten que el alumno aprenda a su propio ritmo y se enfrente de modo individual al proceso de elaboración de sus propias conclusiones con relación a los fenómenos que va a simular, de esta manera el alumno asume la responsabilidad de su propio proceso de aprendizaje.

La autoevaluación: la simulación permite al alumno realizar acciones orientadas a su propia autoevaluación mediante el planteamiento de guiones y cuestionarios orientados al tema que está estudiando.

2. PROPUESTA

El desarrollo de esta propuesta tiene base en las recomendaciones de Blanco (2007), que señala que para hacer planeaciones basadas en competencias es importante considerar algunos aspectos como: la contextualización de la asignatura en el proyecto educativo general, el número y tipo de competencias a formular para una asignatura y los elementos de un sistema de evaluación basado en competencias.

En ese sentido, de acuerdo con Blanco (2007), se reconoce que hay que realizar las siguientes acciones para programar los cursos por competencias:

1. Definir las competencias del módulo (curso) en términos de resultados de aprendizaje.
2. Diseño de una evaluación auténtica (rúbrica de evaluación).
3. Diseño de actividades de aprendizaje.
4. Implementación de la evaluación.

Al seguir estas recomendaciones, en primer lugar se define el contexto de la UA en el Plan de Estudios de la licenciatura, sin omitir su contexto institucional. En este sentido es necesario considerar que todos los planes de estudio de la UAEM se desarrollan en el marco del modelo de Innovación curricular promovido en la Universidad desde 2004, caracterizado por establecer una estructura sistémica del proceso formativo en el nivel profesional, la orientación de la formación profesional basada en competencias, el modelo de enseñanza y aprendizaje innovador, la estructura curricular basada en núcleos de formación, actos académicos expresados en créditos y un sistema de seguimiento y evaluación (Moreno, Medina, Espinoza y Miranda, 2005). En consecuencia, los planes de estudio de la carrera de Ingeniería en Computación tienen la finalidad de promover la participación activa del alumno en la construcción de su conocimiento y la acción mediadora del docente que promueva la evaluación formativa que retroalimente la metacognición en el alumno.

Acorde con los lineamientos institucionales, esta propuesta se diseñó considerando que Programación Estructurada se imparte en el primer semestre de la licenciatura, durante tres horas/semana de teoría y tres horas/semana de prácticas, y representa un valor de nueve créditos curriculares; corresponde al núcleo básico, el cual es fundamental en la comprensión la inserción de la profesión en un complejo disciplinario, interdisciplinario y transdisciplinario, en él de desarrollan competencias básicas para la atención de temas transversales. (Moreno y cols. 2005).

Es así que la Programación Estructurada es la UA en la que se introduce al alumno al conocimiento del paradigma de la programación estructurada y de la programación modular, sin que el alumno tenga conocimientos previos de la disciplina. En esta UA el alumno desarrolla la lógica de programación necesaria para resolver problemas de dificultad media y adquiere las bases teórico-metodológicas para comprender otros paradigmas de programación.

Definición de las competencias a desarrollar: de acuerdo con el plan de estudios aprobado en el año 2009 por los H. Consejos de Gobierno y Académicos de la Facultad de Ingeniería de la UAEM, en la UA se desarrollan cinco Unidades de competencia, las cuales son:

- Identificar las fases de la metodología de programación estructurada para la solución de problemas.
- Aplicar la programación estructurada en la solución de problemas utilizando lenguaje informal y diagramas de flujo.
- Utilizar arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas.
- Usar técnicas de programación modular en el desarrollo de programas informáticos.
- Utilizar los registros para almacenar y manipular información en el desarrollo de programas informáticos.

En opinión de Castro (n. d.) “para facilitar el aprendizaje de la programación es necesario presentar la programación como un nuevo lenguaje de comunicación, hay que hacer énfasis en los elementos del lenguaje como la sintaxis, la semántica, el alfabeto, las expresiones y sentencias y cómo estos se articulan para poder escribir programas, además es preciso un sistema de evaluación que permita conocer si el estudiante está capacitado para construir algoritmos”, atendiendo a esta sugerencia, a continuación se definen las competencias en términos de resultados de aprendizaje.

Identificar las fases de la metodología de programación estructurada para la solución de problemas		
Resultados de Aprendizaje	Conocimientos	Fases de la solución de un programa usando la computadora
		Sentencias, identificadores, expresiones, reglas de precedencia de signos operacionales, estructuras de datos
		Estructuras de control Algoritmo y su representación
	Habilidades (saber hacer)	Analizar las fases de la solución de problemas de programación estructurada.
		Diseñar diferentes soluciones para resolver problemas informáticos
Aplicar la programación estructurada en la solución de problemas utilizando lenguaje informal y diagramas de flujo		
Resultados de Aprendizaje	Conocimientos	Caracterización de las diferentes estructuras de datos
		Caracterización de las diferentes estructuras de control.
		Las instrucciones básicas
		Las características de un diagrama de flujo
		Las características de un pseudocódigo
		Los pasos para realizar una prueba de escritorio
	Habilidades (saber hacer)	Identificar los tipos de problemas que se abordan con las distintas estructuras de control
		Solucionar problemas utilizando estructuras de control
		Utilizar instrucciones básicas en la elaboración de programas.
		Diseñar programas utilizando diagramas de flujo.
		Diseñar programas utilizando pseudocódigo.
		Realizar la prueba de escritorio de un programa
		Diseñar creativamente las diferentes soluciones de un programa informático
		Codificar y evaluar los algoritmos propuestos como solución a los problemas
Utilizar arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas		
Resultados de Aprendizaje	Conocimientos	Las características de las estructuras de datos estáticas
		El manejo de los Arreglos unidimensionales
		Las características y el manejo de los Arreglos bidimensionales y multidimensionales
	Habilidades (saber hacer)	Reconocer los tipos de problemas o situaciones que pueden resolverse usando los arreglos
		Diseñar programas utilizando las estructuras de datos de tipo arreglo.
		Diseñar creativamente las diferentes soluciones de un programa informático
		Codificar y evaluar los algoritmos propuestos que incluya el manejo de arreglos

Usar las técnicas de programación modular en el desarrollo de programas informáticos		
Resultados de Aprendizaje	Conocimientos	Las variables globales y locales
		Las propiedades de los argumentos y parámetros formales
		Las características principales de la programación modular
	Habilidades (saber hacer)	La declaración y la llamada a un módulo
		Diseñar programas utilizando variables globales y locales en la realización de un programa modular
		Analizar y Diseñar creativamente las diferentes soluciones de un programa informático
		Codificar y evaluar los algoritmos propuestos con la técnica de divide y vencerás
Resultados de Aprendizaje		
Resultados de Aprendizaje	Conocimientos	Las características de las estructuras de datos de tipo registro
		Los registro simples
		El registro de registro
		Los arreglos de registros
	Habilidades (saber hacer)	Diseñar programas utilizando las estructuras de datos de tipo registro.
		Diseñar diferentes soluciones de un programa informático

Tabla 1: Competencias en términos de resultados de aprendizaje

Las actitudes se manifiestan con la predisposición de actuar positiva o negativamente en el proceso de aprendizaje, estas actitudes son variables en el tiempo por eso se propone promover y fomentar las actitudes positivas en el grupo de alumnos. Las actitudes a observar durante el aprendizaje son: el respeto a los compañeros y a sus opiniones, la perseverancia, la tenacidad, la constancia y la dedicación.

Diseño de una evaluación auténtica: la evaluación tiene como propósito medir las competencias, mediante la observación de las capacidades y actitudes del estudiante, a partir de indicadores que buscan determinar el grado de desarrollo de las competencias en los ámbitos cognitivo y afectivo, para brindar retroalimentación en torno a fortalezas y aspectos que es necesario mejorar (DGEBR, n, d). Es un elemento muy importante del proceso de aprendizaje, por medio de la valoración se pueden comprender los resultados del proceso de aprendizaje, en palabras de De León (n, d), "A través de la evaluación se obtiene información que orienta la toma de decisiones para mantener y mejorar lo que está bien y para resolver lo que está mal."

Esta propuesta se propone evaluar con un enfoque por competencias, que en opinión de Cano (2008), se pueden sintetizar en integrar conocimientos, habilidades y actitudes, realizar ejecuciones, actuar de forma contextual, entender el proceso de forma dinámica (adquisición de la competencia de forma gradual) y actuar con autonomía. La evaluación propuesta está orientada a recabar información sobre el proceso del aprendizaje, considerando su naturaleza dinámica y sus efectos observables.

En el diseño de los instrumentos de evaluación se parte de que las competencias no pueden ser observadas directamente, sino que se infiere a partir del desempeño y de los productos concretos que resultan (De León, n. d.), por lo tanto es necesario definir las evidencias esperadas en el proceso de aprendizaje y su correspondiente instrumento de evaluación, que se apliquen continuamente durante el proceso de aprendizaje con la intención de monitorear el resultado de las actividades escolares y su contribución al logro de los objetivos, en este caso se proponen las rúbricas, pero es posible también usar listas de cotejo o guías de observación.

En la tabla 2 se describe el diseño de algunas rúbricas propuestas para evaluar los productos generados de las actividades de aprendizaje.

Con la sumatoria de las rúbricas se pueden definir puntajes para identificar cuatro grados del desarrollo de competencias:

- Grado 1: No se esfuerza por adquirir la competencia y no demuestra haberla adquirido o lo hace rara vez.
- Grado 2: Estudia, se forma y demuestra que aplica algunas veces la competencia.
- Grado 3: Ha aprendido la competencia y en su conducta se demuestra que la aplica.
- Grado 4: Tiene integrada la competencia en sus hábitos de conducta.

Aplicar la programación estructurada en la solución de problemas utilizando lenguaje informal y diagramas de flujo			
ACTIVIDADES	PRODUCTOS	RÚBRICA	
El alumno diseñará algoritmos que resuelvan problemas propuestos por el docente en acetatos para explicarlos a sus compañeros en el aula y en salas de computo usarán el programa DFD para analizar el estado de las variables y el comportamiento de su algoritmo.	Algoritmos de problemas donde pueda observarse el uso de las estructuras de control.	1. Se hace uso adecuado de los símbolos del DFD	2
		2. El algoritmo cumple con todas sus características (finito, preciso, exacto)	2
		3. El alumno explica con palabras adecuadas el algoritmo	2
		4. El alumno considera flujos alternativos	2
		5. El algoritmo resuelve el problema	2
		Puntos en total	10
Utilizar arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas			
ACTIVIDADES	PRODUCTOS	RÚBRICA	
Codificar los algoritmos previamente diseñados con ayuda de Dev C y validar los resultados	Algoritmos y códigos en lenguaje C generados a partir de problemas propuestos	1. Se hace uso adecuado de la sintaxis del lenguaje	2
		2. El código corresponde con la especificación del algoritmo	2
		3. El alumno explica con palabras adecuadas el código	2
		4. Se usan comentarios como estrategia de documentación interna	2
		5. El código presenta resultados válidos	2
		Puntos en total	10
ACTIVIDADES	PRODUCTOS	RÚBRICA	
Los alumnos codifican en C algunos algoritmos para resolver problemas cotidianos a la empresa, usando funciones	Algoritmos y códigos de subrutinas que interactúen en la solución de problemas	1. Se hace uso adecuado de la sintaxis para definir funciones	2
		2. El código corresponde con la especificación del algoritmo	2
		3. El alumno explica adecuadamente la división de tareas	2
		4. Se usan de forma adecuada los argumentos y valores de retorno	2
		5. El código presenta resultados válidos	2
		Puntos en total	10

Tabla 2: Ejemplos de rúbricas para evaluar las evidencias de aprendizaje

Para evaluar las actitudes, también se elaboran indicadores relativos a las predisposiciones del estudiante. Los indicadores que podemos formular se organizan en una lista de cotejo como el siguiente ejemplo:

Actitud	Indicador	SI	NO
Respeto las diferencias de opinión de sus compañeros	Trabaja en grupos de trabajo en los que se expresan las opiniones libremente		
Tenacidad y dedicación	Demuestra trabajo completo en sus prácticas y ejercicios		
Perseverancia	Demuestra perseverancia ante las dificultades de las actividades realizadas		
Constancia	Presenta todas las actividades a realizar (tarear, prácticas, investigaciones)		

Tabla 3: Ejemplos de Lista de cotejo para evaluar actitudes

Durante el proceso de aprendizaje también es importante que se practique la autoevaluación. La autoevaluación es un recurso que requiere una mirada autocrítica, así como el reconocimiento de actitudes y comportamientos asociados a diferentes temas. Es un proceso de evaluación personal que puede ser socializado. Los beneficios de la autoevaluación son variados, pues permite que los alumnos tomen conciencia por sí mismos de sus logros, de sus errores e identifiquen los aspectos que pueden mejorar durante el aprendizaje. Esto genera en cada estudiante una actitud más responsable ante su propio aprendizaje.

Con la autoevaluación se construye la autonomía en el estudiante, el autoconocimiento, y la autorregulación, es decir, conociendo qué competencia debe desarrollar y cómo hacerlo, la autoevaluación podrá llevar al alumno a la acción sistemática de la metacognición.

Con la finalidad de practicar continuamente la autoevaluación, en esta propuesta se recomienda iniciar la semana con una descripción de las actividades a realizar y los aprendizajes que se lograrán, con esta información se inicia el llenado de un cuadro C-Q-A, el cual posteriormente será finalizado una vez que las actividades se hayan concluido, de esta forma se pretende que el alumno realice una reflexión sobre su desempeño y logros de aprendizaje.

Condernarán (n, d) recomienda llenar los cuadros C-Q-A con el planteamiento de una serie de preguntas antes de la actividad y después de ella contestar dichas cuestiones, de tal forma el alumno descubra lo aprendido con la actividad, y de no lograr resolver todas las dudas, observa la necesidad de seguir investigando y conociendo sobre el tema.

C ¿Qué conozco del tema?	Q ¿Qué quiero aprender del tema?	A ¿Qué aprendí del tema?
Se sugiere que el alumno identifique conocimientos anteriores relacionados con la nueva temática.	Se sugiere que se plantee preguntas sobre el nuevo tema.	Se sugiere que el alumno responda a las preguntas de la sección Q e identifique el conocimiento adquirido y la necesidad de seguir conociendo sobre el tema pues será posible que no todas las preguntas hayan sido resueltas.

Tabla 4: Cuadros C-Q-A

Finalmente, para la evaluación sumativa, se requiere obtener un indicador numérico que represente el aprendizaje adquirido durante el curso; se propone una matriz de evaluación pues, es un instrumento útil para planificar la evaluación y dar explicaciones objetivas sobre los resultados del aprendizaje y el porqué de las calificaciones de los estudiantes.

En este caso se propone una matriz basada en las sumatorias de las rúbricas para las actividades de aprendizaje, los puntajes serán compilados durante el transcurso del semestre.

Identificar las fases de la metodología de programación estructurada para la solución de problemas		P. Evaluación	P. Autoevaluación
Conocimientos	Fases de la solución de un programa usando la computadora		
	Sentencias, identificadores, expresiones, reglas de precedencia de signos operacionales, estructuras de datos		
	Estructuras de control Algoritmo y su representación		
Habilidades	Analizar las fases de la solución de problemas de programación estructurada.		
	Diseñar diferentes soluciones para resolver problemas informáticos		
Aplicar la programación estructurada en la solución de problemas utilizando lenguaje informal y diagramas de flujo			
Conocimientos	Caracterizar las diferentes estructuras de datos		
	Caracterizar las diferentes estructuras de control.		
	Conocer las instrucciones básicas		
	Describir las características de un diagrama de flujo		
	Describir las características de un pseudocódigo		
	Describir los pasos para realizar una prueba de escritorio		
Habilidades (saber hacer)	Elaborar programas que utilizan estructuras de datos		
	Solucionar problemas utilizando estructuras de control		
	Utilizar instrucciones básicas en la elaboración de programas		
	Diseñar programas utilizando diagramas de flujo.		
	Diseñar programas utilizando pseudocódigo.		
	Realizar la prueba de escritorio de un programa		
	Diseñar creativamente las diferentes soluciones de un programa informático		

Tabla 5: Sección de la Matriz de evaluación

Diseño de actividades de aprendizaje: con la finalidad de desarrollar las habilidades y obtener los resultados de aprendizaje que se determinan en el plan de estudios por competencias, se proponen actividades que permitan al alumno comprender los conceptos teóricos y llevar esos conocimientos al nivel de aplicación. Por esta razón se recomienda aprovechar las herramientas computacionales existentes en la actualidad, que ayuden al alumno a comprender la relación de cada concepto teórico con el ciclo de vida de un programa.

Actualmente existe software que permite el diseño de algoritmos, como Visio, que es una aplicación incluida en el paquete de ofimática Office de Microsoft, con costo de licencia; esta herramienta tiene la limitante de solo permitir el dibujo de los distintos símbolos de un diagrama de flujo pero no ofrece otros servicios útiles para la enseñanza de la programación como la ejecución animada del algoritmo y la visualización del manejo de la memoria.

DFD es un simulador gratuito que actualmente puede ejecutarse en Windows 7 y que tiene la propiedad de facilitar el diseño de algoritmos, incluyendo subprogramas y además muestra la ejecución del algoritmo en modo normal o paso a paso. También permite visualizar el estado de las variables involucradas en un algoritmo, lo que facilita al alumno comprender el comportamiento del flujo del programa en cada una de las estructuras de programación y la importancia del orden en la secuencia de pasos.

Otra herramienta útil para explicar al alumno el comportamiento interno del estado de la memoria y su relación con las sentencias del programa es Jeliot. Esta aplicación es un simulador desarrollado en Java y para Java, tiene la ventaja de mostrar con animación el comportamiento interno de los registros y variables en la memoria principal y su cambio con cada línea de código. Aunque está pensada como apoyo para el aprendizaje de la programación orientada a objetos, puede usarse desde unidades de aprendizaje anteriores sobre todo para visualizar aspectos como la asignación de valores por teclado, la evaluación de las condicionales en los ciclos repetitivos y la recursividad entre otros temas.

En la imagen 1 puede observarse cómo Jeliot por medio de animación visualiza el estado de las variables en un programa que llama recursivamente a una función. La visualización del comportamiento del programa al interior de la RAM, facilita al alumno comprender tanto el manejo de memoria como el funcionamiento de cada una de las sentencias de su código.

Un tema que frecuentemente se presenta complejo para la comprensión de los alumnos es el manejo de arreglos, tanto unidimensionales como bidimensionales. Con la animación de la lectura y manejo de los arreglos con Jeliot el alumno puede comprender mejor el papel que juega la variable usada para recorrer el índice del arreglo y las variables usadas para manipular el contenido de las distintas localidades del arreglo. Estas herramientas son particularmente útiles para alumnos con estilos de aprendizaje visuales y kinestésicos

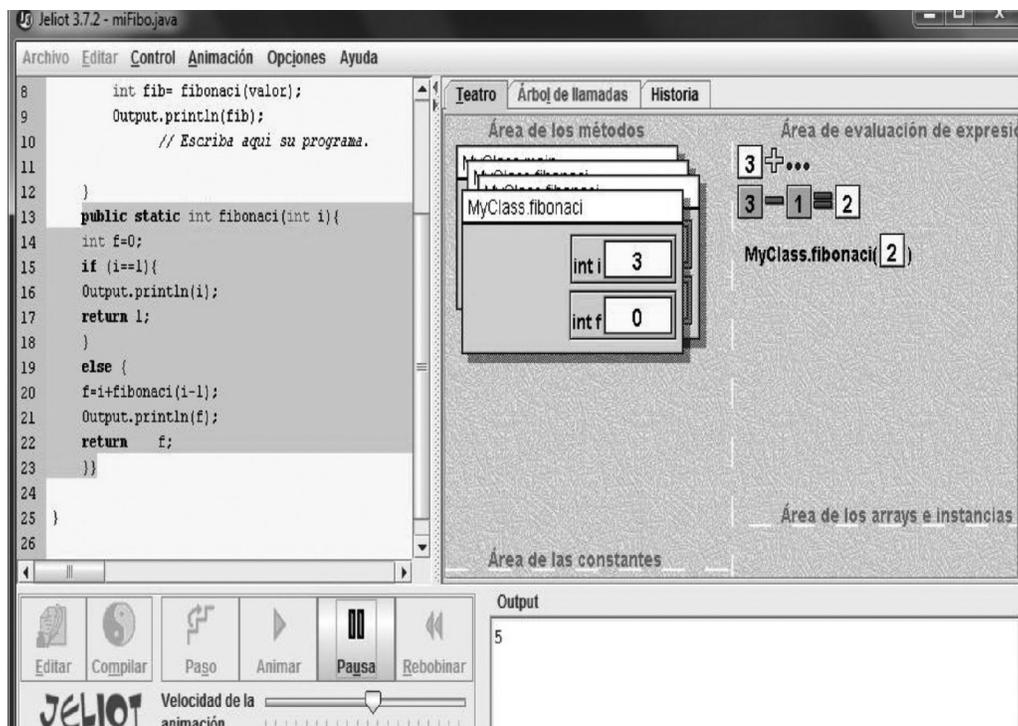


Imagen 1: Comportamiento de la memoria principal en la recursividad con Jeliot

En la tabla 4 se describe con mayor detalle el conjunto de actividades propuestas para recuperar los conocimientos previos de los alumnos y vincularlos progresivamente con conceptos más abstractos de la programación estructurada, en ella se proponen actividades basadas en el uso de simuladores como DFD y Jeliot como recursos didácticos para que el alumno observe mediante las animaciones del software el comportamiento de la memoria, la asignación de valores en las variables y los efectos de las sentencias de control en el algoritmo.

Identificar las fases de la metodología de programación estructurada para la solución de problemas		
COMPETENCIAS A DESARROLLAR	ACTIVIDADES SUGERIDAS	RECURSOS
<ul style="list-style-type: none"> Saber: tipos de problemas, conceptos como metodología de solución de problemas, computadora, elementos y propiedades de la computadora, programa, lenguajes de programación, tipos de lenguajes de programación, compilador, entorno de desarrollo, datos, sentencias, estructuras de programación, variables, identificadores. Hacer: identificar problemas que se resuelvan con ayuda de la computadora. Explicar la posible solución a un problema Ser: respetuoso con los compañeros, paciente, ordenado. 	<ul style="list-style-type: none"> Iniciar el trabajo lluvias de ideas sobre lo que entienden sobre el concepto de instructivo, receta de cocina y manuales y sus características. (es necesario hacer un vínculo entre los conocimientos que el alumno ya tiene para contextualizar los nuevos conocimientos). Explicar a los jóvenes los pasos necesarios para resolver problemas con ayuda de la computadora y ejemplificar dichos pasos. En una mesa de debate los alumnos participan con ejercicios de identificación de problemas de la vida cotidiana que se puedan resolver con la metodología analizada. 	<ul style="list-style-type: none"> Pizarrón, Proyectores de Diapositivas, acetatos Instructivos o manuales. Tareas de investigación, Lecturas de libros de texto.
Aplicar la programación estructurada en la solución de problemas utilizando lenguaje informal y diagramas de flujo		
COMPETENCIAS A DESARROLLAR	ACTIVIDADES SUGERIDAS	RECURSOS
<ul style="list-style-type: none"> Saber: conceptos como sentencias de control, estructuras de programación, algoritmo y su representación, pseudocódigo, diagramas de flujo, variables, corridas de escritorio, palabras reservadas, identificadores, condicionales, operadores aritméticos, lógicos y relacionales. Hacer: El alumno debe ser capaz de especificar un algoritmo con distintas representaciones: diagramas de flujo, pseudocódigo, lenguaje natural. Usará software para diseño de algoritmos como Visio, DF o cualquier editor de imágenes. Ser: Ordenado y paciente, responsable, colaborativo 	<ul style="list-style-type: none"> Por medio de la demostración en el aula, el alumno conocerá las reglas para representar algoritmos con pseudocódigo y diagramas de flujo. El alumno diseñará algoritmos que resuelvan problemas propuestos por el docente en acetatos para explicarlos a sus compañeros en el aula y en salas de cómputo usarán el programa DFD para analizar el estado de las variables y el comportamiento de su algoritmo. Por medio de ejemplos y diapositivas el profesor explica la forma de usar el entorno de programación Dev C, se codifican programas con las estructuras básicas de la programación en C, a partir de los algoritmos previamente desarrollados. El alumno en equipo analizará, diseñará y explicará con tecnicismos el problema y la solución. 	<ul style="list-style-type: none"> Pizarrón, Proyectores de diapositivas o acetatos Problemas a resolver Tareas de investigación, Lecturas de libros de texto. Equipo de cómputo con DFD instalado
Utilizar arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas		
COMPETENCIAS A DESARROLLAR	ACTIVIDADES SUGERIDAS	RECURSOS

<ul style="list-style-type: none"> • Saber: estructuras de datos estáticas y sus propiedades, arreglos, índices, iteraciones, matrices, arreglos multidimensionales. • Hacer: identificar problemas que se resuelvan con el uso de estructuras estáticas (arreglos), almacenar en matrices y vectores, recorrer las localidades en la memoria para almacenar, procesar y visualizar datos. • Ser: perseverante y dedicado 	<ul style="list-style-type: none"> • Investigar sobre las variables definidas por el programador (arreglos unidimensionales y bidimensionales) y analizan la similitud y diferencia entre el manejo de las variables que ya conocen con los arreglos. • Diseñar algoritmos en DFD para resolver problemas cotidianos a la empresa, usando arreglos. • Realizar análisis sobre lecturas de textos sobre las reglas para usar arreglos con la sintaxis de C. • Ejercicios de corridas de escritorio usando Jeliot, para comprender el manejo de los arreglos. • Codificar los algoritmos previamente diseñados con ayuda de Dev C y validar los resultados 	<ul style="list-style-type: none"> • Pizarrón, • Proyector de diapositivas o acetatos • Problemas a resolver • Pareas de investigación, • Lecturas de libros de texto. • Equipo de cómputo con DFD, Jeliot y Dev C++
Usar técnicas de programación modular en el desarrollo de programas informáticos		
COMPETENCIAS A DESARROLLAR	ACTIVIDADES SUGERIDAS	RECURSOS
<ul style="list-style-type: none"> • Saber: conceptos como programación modular, funciones, procedimientos, ámbito de las variables, variables locales y globales, argumentos, parámetros, pase de parámetros, valor de retorno. • Hacer: declarar funciones, usar pase de parámetros por valor, llamar a funciones, usar el valor de retorno. • Aplicar la estrategia de “divide y vencerás”, estructurar problemas usando las funciones • Ser: paciente, persistente, ordenado, analítico 	<ul style="list-style-type: none"> • Los alumnos buscan información sobre los siguientes conceptos: programación modular, funciones, procedimientos, ámbito de las variables, variables locales y globales, argumentos, parámetros, pase de parámetros, valor de retorno. • Realizar análisis sobre lecturas de textos sobre las reglas para definir, especificar y usar las funciones con la sintaxis de C. • Ejercicios de corridas de escritorio usando Jeliot, para comprender la llamada a funciones y el pase de parámetros. • Los alumnos codifican, evalúan o corrigen algunos algoritmos para resolver problemas cotidianos a la empresa, usando funciones. 	<ul style="list-style-type: none"> • Pizarrón, • proyectores de diapositivas o acetatos • problemas a resolver • Tareas de investigación, • lecturas de libros de texto. • Equipo de cómputo con DFD, Jeliot y Dev C++
Utilizar los registros para almacenar y manipular información en el desarrollo de programas informáticos		
COMPETENCIAS A DESARROLLAR	ACTIVIDADES SUGERIDAS	RECURSOS
<ul style="list-style-type: none"> • Saber: definición de estructuras, arreglos de registros, registros con arreglos. • Hacer: Realizar programas que permitan almacenar información en registros simples, registros con arreglos y arreglos de registros, con programación modular. • Ser: persistente, autogestor, curioso para proponer varias soluciones a los problemas. 	<ul style="list-style-type: none"> • Los alumnos investigan sobre las variables definidas por el programador (registros) • Ejercicios de análisis de problemas que puedan resolverse con el apoyo de los registros. • Realizar programas que integren las funciones, los arreglos y los registros, con la finalidad de entrenar al alumno en el manejo de estas estructuras de datos. 	<ul style="list-style-type: none"> • Pizarrón, • Proyector de diapositivas o acetatos • Problemas a resolver • Tareas de investigación, • Lecturas de libros de texto. • Equipo de cómputo con DFD, Jeliot y Dev C++

Tabla 6: Diseño de Actividades para desarrollar las competencias

CONCLUSIÓN

En este documento se muestra la utilidad de la simulación y su pertinencia en el aprendizaje de habilidades como la programación de computadoras. La enseñanza basada en el desarrollo de competencias implica la aplicación de estrategias de aprendizaje centradas en el alumno, promoviendo el aprendizaje activo, el desarrollo de la metacognición y el aprendizaje basado en problemas. En consecuencia la evaluación también debe ajustarse al proceso dinámico y gradual de la adquisición de competencias. Para el aprendizaje de la programación ya se cuenta con herramientas de software que por medio de animaciones facilitan al alumno la comprensión del funcionamiento de la computadora por lo tanto es importante proveer al alumno de estas herramientas para facilitarle la comprensión de los conceptos fundamentales de la programación, la especificación de los problemas, la propuesta de soluciones (algoritmos) y la evaluación de las soluciones. Los simuladores aquí sugeridos facilitan al alumno comprender cómo funcionan las estructuras de control, las llamadas a funciones y el manejo de memoria, su uso promueve el desarrollo de su abstracción y de su lógica de programación.

BIBLIOGRAFÍA

- ANIEI Asociación Nacional de Instituciones de Educación en Informática (2012). Consultado en noviembre de 2012, disponible en línea <http://aniei.org.mx/portal/modules.php?&name=modeloslic2&op=perfiles2>
- BLANCO BLANCO, Ángeles (2007). *Programación y evaluación por competencias*, Universidad Complutense de Madrid, Consultado en julio 2012, disponible en línea <http://www.xtec.cat/serveis/crp/a8930013/capestudi/noucurri/6univmadrid.pdf>
- CONDEMARÍN, Mabel, (n, d), *Estrategias de enseñanza para activar los esquemas cognitivos de los estudiantes*, consultado en febrero 2013, disponible [en línea] http://www.lecturayvida.fahce.unlp.edu.ar/numeros/a21n2/21_02_CondeMarin.pdf
- CANO García María Elena (2008). *La evaluación por competencias en la educación superior*, Profesorado, Revista de Currículum y Profesorado, consultado en agosto 2012, disponible en línea: <http://www.ugr.es/local/recfpro/rev123COL1.pdf>
- CASTRO ORDÓÑEZ, Leví Astul, (n,d). *Competencias Específicas para la enseñanza- aprendizaje de la programación*. Universidad Pedagógica Nacional Francisco Morazán, consultado en agosto de 2012, disponible en línea: http://www.upnfm.edu.hn/bibliod/images/stories/DocDigitales/Paradigmas/paradigma%2014/paradigma14_4to.pdf
- DE LEÓN CERDA Diana, (n, d). *Evaluación integral de competencias en ambientes virtuales de aprendizaje*. Una aproximación a la evaluación del aprendizaje en UDG Virtual". Consultado en agosto 2012, disponible en línea: : http://www.profords.cfe.ipn.mx/profordems3ra/modulos/mod2/pdf/Unidad3/Lecturas/Activ%2016/003_DeLeon.pdf
- DEL PINO GONZÁLEZ, Luis M. *Realidad virtual*. España: Paraninfo, 1995. 210 pp. Consultado en el año 2012.
- DEI, Departamento de Estadística Institucional, UAA, 2009, *Relación por Centro de las cinco materias más reprobadas durante los últimos cinco semestres, entre 2006 y 2008*, consultado en agosto de 2012. Disponible en línea: <http://www.desdelared.com.mx/2009/universidad/091110-reprobadas-uaa.pdf>
- DIRECCIÓN GENERAL DE EDUCACIÓN BÁSICA REGULAR (DGEBR) Perú (2011), *Orientaciones para la evaluación del aprendizaje*, consultado en febrero 2013. Disponible en línea: <http://ebr.minedu.gob.pe/des/pdfs/personal/personaevaluacion.pdf>
- LÓPEZ DE MUNAIN, Claudia Victoria, y Claudia Alejandra Saiegg (2005), *Uso de la simulación como estrategia de mejora en el proceso de enseñanza-aprendizaje en las universidades. Una aplicación para la carrera de Informática*, consultado en febrero de 2013. Disponible en línea <http://www.roa.unp.edu.ar:8080/graduate/bitstream/123456789/202/1/Tesina%20Simulacion%20en%20Educacion.pdf>
- MANCIO Reyes, Carlos Antonio. *La realidad virtual aplicada al tratamiento de la claustrofobia*. Guatemala 2007. Consultado en noviembre de 2012, disponible en línea: http://biblioteca.usac.edu.gt/tesis/08/08_0344_CS.pdf
- MARTÍNEZ TREVIÑO, Yolanda (2005). *En busca de una nueva forma de enseñar a programar*. Investigación bibliográfica, Tecnológico de Monterrey, consultado en julio de 2012. Disponible en línea: [http://www.mty.itesm.mx/rectoria/dda/rieee/pdf-05/27\(DTIE\).YolandaMtz..pdf](http://www.mty.itesm.mx/rectoria/dda/rieee/pdf-05/27(DTIE).YolandaMtz..pdf)
- MICHEL NAVA, Rosa María (2011) *MCCLumin como apoyo para disminuir los índices de reprobación*, consultado en septiembre de 2012. Disponible en línea <http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/3199/MCC%20Lumin.pdf?sequence=1>

- MONGE NÁJERA, Julián, y Víctor Hugo Méndez Estrada (2007). *Ventajas y desventajas de usar laboratorios virtuales en educación a distancia: la opinión del estudiantado en un proyecto de seis años de duración*. Educación, 91-108.
- MORA REYES, Raúl; Adriana Pérez López, y Naty Rodríguez Ventura (2010). *Desarrollo de herramienta software para el apoyo en la comprensión de estructuras básicas de programación*, Tesis para obtener el grado de Maestría en Sistemas Computacionales de la Universidad Popular Autónoma del Estado de Puebla, consultado en agosto de 2012. Disponible en línea: http://cecip.upaep.mx/titulacion_sep/docs/LI2EJ1.pdf
- MORENO ZAGAL, Maricruz; Lourdes Medina Cuevas, Dora Elena Espinoza Angulo, y David Miranda García (2005). *Bases para el modelo de innovación curricular*. Universidad Autónoma del Estado de México, Secretaría de docencia, consultado en agosto de 2012. Disponible en línea.
- RUIZ GUTIÉRREZ, José Manuel, (n,d), *La Simulación como Instrumento de Aprendizaje (Evaluación de Herramientas y Estrategias de Aplicación en el Aula)*. Consultado en febrero de 2013. Disponible en línea: <http://mami.uclm.es/jmruiz/materiales/Documentos/simulacion.PDF>
- SATORRE CUERDA, Rosana; Faraón Llorens Largo, y Juan Antonio Puchol García (1996). *Universidad de Alicante*, consultado en agosto de 2012. Disponible en línea: <http://www.dccia.ua.es/~faraon/docs/programacion.pdf>
- VILLALOBOS SALCEDO, Jorge Alberto, 2009, *Proyecto Cupi2 Una solución Integral al problema de enseñar y aprender a programar*; Universidad de Los Andes, consultado en Agosto 2012, disponible On-Line: http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-205832_recurso_1.pdf