



**UNIVERSIDAD MICHOACANA
DE SAN NICOLÁS DE HIDALGO**



Facultad de Ciencias Físico-Matemáticas

“Mat. Luis Manuel Rivera Gutiérrez ”

**Enseñanza de principios básicos de programación
utilizando realidad aumentada**

Tesis

que para obtener el título de:

Licenciado en Ciencias Físico-Matemáticas

Presenta:

Maria Isabel Lopez Huerta

Director de Tesis:

Mc. Cuauhtémoc Rivera Loaiza

Morelia, Michoacán. Agosto 2019

DEDICATORIA

A mis padres, por hacer que mi hermano y yo crecieramos rodeados de pláticas que no logré entender y me dirigieron a estudiar algo para comprender una mínima parte de estas; a todos eso amigos que se convirtieron en mis profesores, en particular a Jose Leonel Torres, por darme las palabras adecuadas cuando más la necesité, recordarme que soy capaz y que solo importaba hacer las cosas que te gustaran, sin importar cuanto demorara.

Agradecimientos

- A mis padres, por siempre apoyarnos, darnos todo lo que esta a su alcance y más, pero antetodo por su amor incondicional.
- A mi padre, Armando López Zamudio, por heredarme su locura, meterme en este mundo indirectamente y demostrar que se vive mejor si haces las cosas con amor y pasión.
- A mi mamá, Inés Huerta Tzintzún, por tolerar cada fin de estrés y respirar hondo cada que la carga de trabajo hablaba y en ocasiones obligarme a tomar un respiro.
- A mi hermano, que muy a su manera me impulsa a que día a día salga de mi zona de confort y me demuestre todo lo que puedo hacer.
- A Francisco Santoyo Valdez, por darme su apoyo incondicional en cada uno de mis proyectos, convertir las semanas de estrés en semanas de trabajo arduo, motivarme, hacerme ver las cosas con claridad cuando era necesario y sobre todo creer en mí.
- A cada una de las personas que me ayudaron a que este trabajo fuera posible, en especial a mi asesor Mc. Cuauhtémoc Rivera Loaiza, por orientarme e impulsarme a colaborar y convertirse en un amigo.

Índice de figuras

2.1.	Interfaz de karel. Foto tomada de Tutorial de Karel el Robot Olimpiada Mexicana de Informática Abril 2015. César Cepeda.	18
2.2.	Interfaz de Scratch. Foto tomada de [1].	18
2.3.	Interfaz de Alice. Foto tomada de [2].	19
3.1.	Descripción del Proceso de un Diseño Centrado en el Usuario. Fuente: Imagen realizada por TektonTech, iconografía freepik.es	22
3.2.	Prueba de usuario haciendo uso de la tecnica de Mago de Oz. Foto tomada de [3]	26
3.3.	Prueba de usuario haciendo uso de la tecnica de Mago de Oz, foto tomada de [3]	27
3.4.	Google Cardboard	29
4.1.	Diagrama de afinidad.	36
4.2.	Explicacion.	38
4.3.	primer prototipo papel.	39
4.4.	primer prototipo papel.	39
4.5.	usuario colocando sus indicaciones.	40
4.6.	prueba.	41
4.7.	primer ejecucion.	42
4.8.	entrevista a usuario.	42

Índice de tablas

2.1. En esta tabla se muestra de forma compacta todas las herramientas a modo de comparación.[4]	16
--	----

Resumen

Se presenta un avance de la propuesta de diseño para un software que permite que personas entre 14 y 18 años puedan tener un acercamiento al mundo de la programación estructurada. Dicho diseño está basado elaborado tomando como base el diseño centrado en el usuario. Se realizó un análisis de herramientas que actualmente existen en el mercado, además de compararlas entre sí, se estudió a fondo los posibles usuarios haciendo entrevistas, focus group y encuestas en un Centro de Bachillerato Tecnológico e industrial y de servicios, donde posteriormente se llevaron a cabo una serie de pruebas de usabilidad en las cuales se empleó la metodología de Mago de Oz.

Palabras clave: Diseño centrado en el usuario, mago de oz, usabilidad, interacción humano-computadora, karel, programación estructurada, realidad virtual.

Abstract

This is a work in progress to approach software design that allow teenagers between 14 and 18 years introduced themselves into structured programming. This design was complete based in the philosophy of User Centered Design. An analysis of the current tools that are in the market was carried out. In addition to comparing each other, we studied the potential users by conducting interviews, focus group and surveys in a Centro de Bachillerato Tecnológico Industrial y de Servicios No. 94, where subsequently a series of usability tests were considered in which the Wizard of Oz methodology was used.

Introducción

Es innegable la importancia que las ciencias de la computación han tenido en todas las áreas del conocimiento humano desde mediados del siglo pasado. Aunado a esto, la ubicuidad de los dispositivos móviles (particularmente los teléfonos inteligentes) que han redefinido las formas de comunicación e interacción con todo el entorno.

En México, de acuerdo con cifras oficiales del Instituto Federal de Telecomunicaciones de México en el 2019 se tenían 79.1 millones de usuarios de internet; 17.4 millones de hogares mexicanos cuenta con una conexión a Internet [5]. Además, los usuarios de computadora de seis años o más alcanzan en 2017 los 50.6 millones, equivalentes al 45.3 % del total de la población en este rango de edad. Todo esto da la indicación de que al menos en las zonas con mayor urbanización el país el acceso a las tecnologías de la información está ampliamente distribuida.

Eso conlleva a la necesidad que como país la producción de material que funcione sobre las plataformas digitales sea una prioridad, ya que de lo contrario se corre el riesgo de depender de otros países para poder operar los distintos sistemas que hacen que un país moderno funcione a cabalidad.

En este sentido, la habilitación y capacitación de la población en la producción de medios digitales debe ser parte de la estrategia de crecimiento y viabilidad de un país. Sin embargo, el lograr tener una población capaz de desarrollar esta clase de productos tiene como componente esencial el aprender a programar, y esta no es una tarea sencilla.

El aprendizaje de un lenguaje de programación conlleva el aprender a estructurar instrucciones en una secuencia lógica que la cual si se ejecuta de manera adecuada llevará a la solución de un problema, y este es un proceso que lleva tiempo y bastante práctica. En esta tesis se considera que la utilización de nuevos enfoques de asimilación de los pasos para programar son necesarios.

El enfoque empleado en este trabajo es el enseñar a poblaciones de adolescentes a estructurar instrucciones simples que conllevan a la solución de un problema. Para esto se utilizó el concepto del Robot Karel[6], el cual ha sido empleado exitosamente para introducir a los estudiantes en el mundo de la programación, pero incorporando una capa de realidad aumentada. Todo esto con la finalidad de que los usuarios logren mediante un proceso inmersivo la adopción de habilidades necesarias para aprender a programar.

Índice general

Índice de figuras	III
Índice de tablas	IV
Resumen	v
Abstract	VI
Introducción	VII
1. Planteamiento del Problema	1
1.1. Descripción del Problema	1
1.2. Objetivo	2
1.2.1. Objetivos Generales	2
1.2.2. Objetivos Específicos	2
1.3. Hipótesis	2
1.4. Justificación	2
1.5. Descripción de la tesis	3
2. Enfoque Pedagógico	4
2.1. Constructivismo	4
2.2. Aprendizaje significativo	6
2.3. El Uso de la Tecnología En la Educacion.	8
2.3.1. Cs Unplugged	9
2.3.2. Videojuegos en el Aula	9
2.3.3. Videojuegos en el Aprendizaje de Programacion Orientada a Obje- tos (OOP)	11
2.4. Comparando Herramientas	12
2.4.1. Karel	14
2.4.2. Scratch	17
2.4.3. Alice	18
3. Diseño Centrado en el Usuario	20
3.1. Metodos de Investigación	22
3.1.1. Observación e investigación contextual	22
3.1.2. Entrevistas	23

3.1.3.	Encuestas	23
3.1.4.	Focus group	24
3.2.	Indagacion al Diseño	24
3.2.1.	Perfil de Usuario	24
3.2.2.	Personas	25
3.3.	Diseño de prototipos y Evaluacion	25
3.3.1.	Prototipado	25
3.3.2.	Tecnica Mago de Oz	26
3.3.3.	Test de Usuario	27
3.4.	Realidad Virtual	28
4.	Descripción del Experimento.	30
4.1.	Observacion e investigacion contextual	30
4.1.1.	Entrevistas	30
4.2.	Personas/ perfil de usuario	32
4.3.	Desarrollo del Diseño Centrado en el Usuario	36
4.4.	Prototipo de Baja Fidelidad	38
4.4.1.	Conclusión de la prueba	39
4.4.2.	Consideraciones a futuro.	40
5.	Conclusiones y Trabajo Futuro	43
A.	Anexo	44
A.1.	Focous Group	44
A.2.	Entrevistas	54
	Bibliografía	59

Capítulo 1

Planteamiento del Problema

Ser capaz de utilizar eficientemente un lenguaje de programación como una herramienta cotidiana es una de las habilidades fundamentales que se deben de tener el mundo actual. Si ésta no se desarrolla desde una edad temprana se corre el riesgo de no poder competir adecuadamente en el mundo laboral que se está viviendo. Sin embargo, se reconoce que en muchos casos no es una tarea sencilla por múltiples factores, por lo que se buscan alternativas de enseñanza para poder incorporar a la población. La propuesta aquí planteada conlleva un proceso de enseñanza de conceptos básicos de programación mediante un enfoque constructivista, utilizando herramientas que son de fácil acceso a la población adolescente de México.

1.1. Descripción del Problema

Las generaciones más jóvenes hoy en día pertenecen al grupo de personas cuya vida ha transcurrido en su totalidad en el ámbito digital. En particular, aquellos que han nacido a partir de los noventas siempre han estado en contacto con un mundo en extremo interconectado y con amplia disponibilidad de dispositivos electrónicos. Sin embargo, esto no significa que sean capaces de producir material para ese ámbito: en la mayoría de los casos se tiene solamente consumidores de tecnología y no productores.

Cada vez es más común pensar que los adolescentes y niños se sienten muy cómodos con la tecnología, pues han ido creciendo con ella, les es sumamente natural enviar mensajes de texto, jugar en línea, hacer búsquedas e incluso generar contenido, etc. Más sin embargo, esto no necesariamente es una indicación de que sean capaces de generar su propio contenido digital; pocos son los que tienen conocimiento en algún lenguaje de programación, algún programa que les permita hacer animaciones o ideas básicas de cómo funcionará la programación que está detrás de estas plataformas o dispositivos que usan diariamente, no mencionemos el intentar leer un código o realizar uno, la influencia digital que tienen no debería limitarse a los casos mencionados al inicio, es importante que también logren comprender al menos tener nociones sobre qué es y cómo se construye un app o alguna otra funcionalidad digital.[7]

Actualmente existen varias herramientas diseñadas para niños y que han demostrado ser muy exitosas a la hora de enseñar a niños a programar, tales como Karel, Alice o scratch [8] [1], por mencionar algunas. Estas herramientas son sumamente llamativas e interesan-

tes para los niños, sin embargo, en el momento que se habla de adolescentes, no tenemos tanta variedad y las herramientas mencionadas no son tan eficaces con ellos como lo son con los niños, esto se debe en gran parte a que los adolescentes están acostumbrados a recibir estímulos visuales todo el tiempo, así que el estímulo que estas herramientas les proporcionan les es insuficiente e incluso les llegan a parecer aburridas o mundanas [9], por ello en esta tesis se plantea el diseño de un videojuego diseñado especialmente para adolescentes que permita desarrollar un conocimiento básico en programación.

1.2. Objetivo

El objetivo principal es implementar un proceso de enseñanza de conceptos básicos de programación mediante la utilización de realidad aumentada. A continuación se exponen las áreas que se pretende estudiar, explorar y contribuir con esta investigación:

1.2.1. Objetivos Generales

Diseñar un software centrado en las necesidades de los adolescentes en el rango de 15 a 18 años de edad, haciendo uso de la realidad aumentada.

1.2.2. Objetivos Específicos

- Analizar las herramientas existentes en el área.
- Empleando metodologías del Diseño Centrado en el Usuario diseñar un sistema de enseñanza de la programación.
- Emplear la Realidad Aumentada para reforzar la adopción de los conocimientos básicos de programación.

1.3. Hipótesis

El uso de la realidad aumentada facilita el aprendizaje de conceptos básicos de programación, mediante el uso de un ambiente computacional simplificado llamado Karel.

1.4. Justificación

Aprender a programar es sumamente importante ya que esto puede llegar a significar la posibilidad de aspirar a mejor oportunidades de empleo o calidad de vida, incluso cuando los estudiantes no quieren desempeñarse en algo remotamente relacionado con el ámbito, les es de mucha utilidad ya que les permite desarrollar su imaginación, pensar de manera más estructurada, trabajar en equipo, desarrolla el razonamiento abstracto, diseño de estrategias, solución de problemas entre otras cosas [1].

1.5. Descripción de la tesis

En este trabajo se presenta un marco teórico el cual está dividido en dos partes, la primera está enfocada a lo pedagógico que fue abordado desde el punto constructivista y el uso de la tecnología en la educación, la segunda parte toma los temas correspondientes a interacción humano computadora, tomando como tema principal el diseño centrado en el usuario.

Durante el capítulo cuatro se expone el trabajo que se realizó al diseñar un software para alumnos de preparatoria, se presentan pruebas de entrevistas, focus group, encuestas y pruebas de usabilidad. El quinto y último capítulo se abordan las conclusiones y trabajo a futuro que podría surgir de este proyecto.

Capítulo 2

Enfoque Pedagógico

Con el paso del tiempo y conforme se han desarrollado nuevas tecnologías, éstas poco a poco se han ido incorporando dentro de las aulas, se han desarrollado técnicas de enseñanza en torno a ellas, e incluso algunos docentes han incorporado tan bien la tecnología que les es indispensable a la hora de impartir algunos temas.

Ahora bien, la inclusión de nuevas tecnologías dentro del ejercicio docente debe de ser aplicada siguiendo buenas prácticas pedagógicas. En particular, nuestra propuesta se basa en la teoría del constructivismo, la cual se menciona a continuación.

2.1. Constructivismo

Existen conocimientos que son transmitidos generación tras generación de forma natural y sin necesidad de un gran esfuerzo, muchos de estos son adquiridos por simple imitación, sin embargo existen muchos otros que requieren un esfuerzo e interés de parte de la persona que pretende construir conocimientos nuevos, para esto el sujeto suele hacer uso de los conocimientos previos esto es muy cercano a lo que actualmente conocemos como la corriente constructivista. [10]

En esta sección se presentan 4 hipótesis extraídas de la psicología genética y la psicología social que apoyan y sientan las bases de esta teoría:

Primera Hipótesis: El aprendizaje se apoya en la acción. Idea fundamental en la obra de Piaget: : «Es de la acción de la que procede el pensamiento en su mecanismo esencial, constituido por el sistema de operaciones lógicas y matemáticas» [11]

Es importante señalar que el término “acción” es utilizado con mucho frecuencia en pedagogía y en didáctica como “llevar a cabo manipulaciones” sobre materiales, en el caso de matemáticas o en el caso particular de este trabajo, no se refiere a la manipulación de objetos físicos o reales. Las “acciones” a las que nos referimos en esta primera hipótesis, si bien pueden tener su origen en manipulaciones reales previas, que podría evocar mentalmente o incluso verbalmente el sujeto, no tienen necesidad de identificarse siempre con manipulaciones efectivas. [10]

Durante la infancia los niños necesariamente inician la construcción de conocimientos a través de acciones concretas y efectivas sobre objetos reales y prueban su validez o invalidez a través de la manipulación de objetos, para poder decir que el individuo a construido un nuevo conocimiento es necesario que parta de un conocimiento previo para llegar a otro, “El conocimiento debe manifestarse como instrumento de decisión anticipada» [12]

Segunda Hipotesis: La adquisición, organización e integración de los conocimientos del alumno pasa por estados transitorios de equilibrio y desequilibrio, en el curso de los cuales los conocimientos anteriores se ponen en duda. [10]

Si un alumno logra sobrepasar este proceso de desequilibrio es ahí cuando se dice que este logra tener un aprendizaje de nuevos conocimientos, según Piaget es un proceso de desorganizar los conocimientos previamente adquiridos y acomodarlo con los nuevos. Chamorro lo explica en su libro “El aprendizaje significativo en el área de matemáticas” [10] de la siguiente manera:

En el curso de la acción sobre un determinado medio, las contradicciones aparecen en el sujeto como producto de los desequilibrios, y debe modificar sus representaciones, se produce lo que Piaget ha denominado acomodación, que supone, básicamente, una modificación en el sujeto causada por el medio (perturbación). De manera recíproca, las transformaciones realizadas por el sujeto para dar respuesta a las perturbaciones, modifican su organización del medio, produciéndose entonces un proceso de asimilación. El doble juego acomodación/asimilación está en el centro de los mecanismos de los procesos de reordenamiento.

Es decir, el aprendizaje rara vez se produce por memorizaciones, a una yuxtaposición de “saber-hacer” o a un condicionamiento, aprendemos raramente de una sola vez; aprender supone volver a empezar, extrañarse, repetir, pero repetir comprendiendo lo que se hace y por qué se hace. [13]

Para poder llegar a reproducir todo este conocimiento es necesario el error, ya que sin él no se enfrentarían a los desequilibrios que llevan al cuestionamiento, autoconvencimiento y debate que orilla a validar o desvalidar las teorías propuestas.

Tercera Hipotesis : Se conoce en contra de los conocimientos anteriores. Se trata de una idea fundamental de la epistemología de Bachelard sobre el conocimiento científico, tomada por Brousseau para explicar la formación de obstáculos en el aprendizaje de las matemáticas: La utilización y la destrucción de los conocimientos precedentes forman parte del acto de aprender [12].

Los aprendizajes previos de los alumnos se deben tener en cuenta para construir nuevos conocimientos, ya que estos no se producen a partir de la nada, su elaboración está sometida a adaptaciones, rupturas y a reestructuraciones, a veces radicales, de los conocimientos anteriores. Aprendemos a partir de y también en contra de lo que ya sabemos. Los nuevos conocimientos no pueden hacerse más que modificando los precedentes y no por la simple acumulación de los últimos sobre los ya existentes. [10]

Cuarta Hipotesis: Los conflictos cognitivos entre miembros de un mismo grupo social pueden facilitar la adquisición de conocimientos. Idea básica de la psicología social apoyada en la obra de (Vygotsky), quien consideraba preciso tener en cuenta lo que un individuo puede hacer con la ayuda de otros, ya que el aprendizaje se produce en un medio social en el que abundan las interacciones, tanto horizontales (niño-niño) como verticales (niño-adulto). [10]

La eficacia de los conflictos sociocognitivos se justifica, según (Blaye 1994), puesto que:

- Permiten al alumno tomar conciencia de otras respuestas diferentes a la suya, lo que le obliga a descentrar su respuesta inicial.
- La necesidad de llevar a cabo regulaciones sociales, para llegar a un consenso, implica que el alumno sea más activo cognitivamente.
- La respuesta diferente de los otros es portadora de información y llama la atención del sujeto sobre aspectos de la tarea que no había considerado.

Así, los conflictos sociocognitivos provocan un doble desequilibrio: “desequilibrio interindividual, debido a las diferentes respuestas de los sujetos; desequilibrio intraindividual, debido a la toma de conciencia de respuestas diferentes, lo que invita al sujeto a dudar de su propia respuesta” (Guilly, 1994).

2.2. Aprendizaje significativo

El aprendizaje significativo, en la labor educativa, ofrece herramientas que no solo sirven para el desarrollo de procesos cognitivos en los estudiantes, también pueden ser un apoyo grande para el diseño de estrategias de enseñanza que sean coherentes con este principio según menciona Moreno y Orozco [14].

Para Ausubel (1983, citado por Moreno y Orozco, 2009) existen tres tipos de aprendizaje significativo : **el aprendizaje significativo de representaciones**, el cual consiste en asociar las palabras a objetos, **el aprendizaje de conceptos**, surge mediante dos procesos el de formación y asimilación, Moreno y Orozco afirman “en la formación de conceptos, las características del concepto se adquieren a través de la experiencia directa del estudiante en sucesivas etapas de formulación y prueba de hipótesis”, por ejemplo el aprendizaje de conceptos por asimilación se tiene cuando un niño comienza a ampliar su vocabulario, **el aprendizaje de proposiciones** “consiste en captar el significado de nuevas ideas expresadas en dos o más términos conceptuales unidas por palabras enlace para producir una unidad semántica” (p.12). Bien no es tan simple conseguir este tipo de aprendizaje, ya que según Ausubel (1983 citado por Moreno y Orozco, 2009), es necesario que se cumplan algunas condiciones. El contenido que desea que el estudiante aprenda debe **presentarse de manera organizada y clara**, de esta forma facilita el procesos de asociación y relación con el conocimiento previo. Es sumamente importante que **el estudiante esté motivado, preparado y dispuesto a esforzarse** para recibir el conocimiento al cual se le está retando a aprender. Necesita **una estructura de conocimientos previos activada y dispuesta** donde poder relacionar los nuevos conocimientos.

En cambio para Coll [15], hablar de aprendizaje significativo es, ante todo, “privilegiar el proceso de construcción de significados como eje vertebral del proceso de enseñanza/aprendizaje”. Es decir los estudiantes solo obtienen un aprendizaje significativo cuando logran asociar el objeto de aprendizaje a un significado. El objeto de aprendizaje puede ser cualquier contenido, norma, procedimiento, valor etc.

Coll cuestiona qué es lo que significa la afirmación “los estudiantes construyen significados”. Coll, Ormerod, Moreno y Orozco tratan de responder este cuestionamiento basándose en algunas aproximaciones de Ausubel para realizar una construcción de significados es necesario el establecimiento de relaciones no arbitrarias entre lo que ya se conoce y lo que se está aprendiendo [15].

Menciona lo siguiente “hay que cambiar la forma de pensar de los maestros con respecto a este punto para que, en lugar que se proponga de manera general que los estudiantes realicen aprendizajes significativos, se proponga mejor que los estudiantes desarrollen aprendizajes significativos en cada grado de escolaridad” (p.134). En este sentido, el autor indica que “las personas construyen significado cuando integran o asimilan el nuevo objeto de aprendizaje a los esquemas que ya poseen de la comprensión de la realidad” (p. 135). De esta manera lo que plantea Coll que cree prudente se debería seguir actualmente, lo cual parece bastante pertinente puesto que los estudiantes necesitan ser dotados de herramientas especializadas, que les permitan desarrollar, ampliar y ahondar en los conocimientos que van o ya adquirieron.

La construcción de significado para el aprendizaje significativo vista desde el punto piagetiano, básicamente es acomodar y asimilar[15] lo cual da entender que cuando se inserta un nuevo conocimiento previo lo imprimimos de significado. De esta manera la construcción de significados “implica igualmente una acomodación, una diversificación, un enriquecimiento, una mayor interconexión de los esquemas previos” [15].

Ahora bien es natural preguntarnos: ¿ el aprendizaje siempre es significativo? pues según lo analizado anteriormente a través de los autores ya citados ¡no! el aprendizaje no siempre es significativo incluso Coll menciona: “en muchas ocasiones el aprendizaje es simplemente mera repetición memorística ya que, en el nivel escolar, es bien difícil alcanzar un nivel elevado de significatividad puesto que las condiciones que exige su realización no siempre son fáciles de cumplir” [15].

Ausubel menciona ciertas condiciones necesarias para que el aprendizaje significativo ocurra será natural que Coll tenga las suyas desde su perspectiva, el menciona que es necesario que los materiales de aprendizaje necesariamente deben tener la capacidad de activar en los estudiantes la construcción de significados; **el contenido que el estudiante va a aprender ha de ser potencialmente significativo**[16]. Lo cual implica que el material a aprender debe de ser significativo por sí mismo, **tener una estructura lógica bien pensada**. Si el contenido a aprender carece de significado lógico, de estructura y no se tiene en cuenta las necesidades y conocimientos previos de los alumnos estos jamás podrán llegar a aprender de manera significativa, es importante que los materiales además de ser significativos lógicamente también lo sean psicológicamente para los estudiantes ya que esto facilitará la recepción del mismo, en este punto es cuando los conocimientos previos del estudiante y el trabajo del docente se combinan y tienen juntos que enfrentar los retos que se presentan al momento de adquirir nuevos conocimientos, estos varían de acuerdo a la metodología que el docente decida seguir, el tipo del grupo al que se enfrenta entre

otras cosas.

Una segunda instancia, tiene que ver con la necesidad de todo estudiante de adoptar una actitud favorable para relacionar los conocimientos adquiridos previamente con los significados ya construidos [15]. Si el nivel de aptitud del estudiante hacia el aprendizaje significativo es insuficiente, la actividad de aprender, con seguridad, no será otra cosa que memorística y repetitiva. Por el contrario, si el nivel de aptitud es superior, la cantidad y variedad de las relaciones que establecerá entre lo nuevo que aprende y sus presaberes probablemente también serán de nivel superior.[16]

2.3. El Uso de la Tecnología En la Educación.

Una de las razones primordiales por las cuales surge un avance tecnológico es hacer más simples las tareas cotidianas, por ello sería natural que en algún momento se llegara a cuestionar ¿cómo es que la tecnología facilita el aprendizaje? o bien, ¿Qué ventajas ha traído al aula?, ¿Qué potencial tiene la tecnología en la educación? esto mismo lo cuestiona Santos en su artículo presentado en el CIEM del 2007, el menciona: “El software dinámico resulta una herramienta útil en la construcción de representaciones “exactas” de entidades geométricas (puntos, segmentos, rectas, círculos, polígonos, medianas, etc.) y permiten visualizar de manera precisa el comportamiento de partes de cierta configuración o representación del problema “[17] ... “el empleo del software puede funcionar como una herramienta de gran utilidad para que los estudiantes participen en procesos de búsqueda y formulación de conjeturas o relaciones y argumentos o justificaciones matemáticas” [18]. Este es un ejemplo de cómo la tecnología puede ser empleada en el aula, durante esta sección se proporcionan algunos ejemplos más de casos de éxito en el uso de herramientas tecnológicas dentro de las aulas.

Santos [17] menciona que el avance en cuanto herramientas tecnológicas se trataba ya era notable y ofrece diversas oportunidades para incorporarlas en el aprendizaje, además de que Kaput en 1992 en una revisión del impacto del uso de la tecnología en la construcción del conocimiento matemático afirmó que “las limitaciones mayores del uso de la computadora en las siguientes décadas serían probablemente menos debidas a las limitaciones tecnológicas y más a las limitaciones de la imaginación humana y a las restricciones [producidas por] de los viejos hábitos y estructuras sociales”.

La existencia de una variedad de herramientas tecnológicas con distintos potenciales para ser utilizadas en la instrucción de diversas áreas de la educación, plantea un reto no sólo a los profesores sino también a los investigadores en educación en términos de ofrecer información sustentada acerca de cómo utilizar esas herramientas en el desarrollo del pensamiento de los estudiantes. Es decir, resulta importante conocer el potencial o ventajas reales que puede ofrecer el uso de determinada herramienta en la construcción del conocimiento de los estudiantes.

“El empleo sistemático de algunas herramientas puede ayudar a los estudiantes a utilizar representaciones de objetos que faciliten la búsqueda de relaciones. Para conseguir esto, es necesario que los profesores conozcan el potencial de esas herramientas y sean capaces de identificar diferentes estrategias que les permitan utilizarlas en sus prácticas de enseñanza [19]” , el uso de herramientas tecnológicas según Camacho y Santos en un

artículo publicado en el 2004 [20] ofrece la oportunidad de que el estudiante conecte con distintos contenidos y busque nuevas relaciones o resultados.

2.3.1. Cs Unplugged

CS Unplugged [21] es un proyecto llevado a cabo por el Computer Science Education Research Group de la Universidad de Canterbury, Nueva Zelanda. El cual tiene como objetivo principal promocionar la informática (Computación en general) como una disciplina interesante, fascinante e intelectualmente estimulante.

Las siglas CS se refieren a “Computer Science”, esta área se tiende a asociar con habilidades o destrezas que involucran la manipulación de la tecnología y herramientas que esta nos proporciona, las actividades de CS unplugged tratan de enseñar a alumnos de 5 a 15 años que es lo que sucede “dentro” de una computadora, de una forma simple y poco convencional, sin necesidad de que estos tengan un gran conocimiento de programación, la mayoría de sus actividades son al aire libre y sin el uso de ningún dispositivo electrónico, sus actividades constan básicamente de que los participantes hagan las pruebas de escritorio de algún algoritmo básico y posteriormente analicen y razonen cómo y por qué funciona dicho algoritmo que acaban de probar, ellos no aprenden cómo programar ni diseñar lo que acaban de probar sin embargo logran comprender el razonamiento que sigue a grandes rasgos lo cual es bastante bueno cuando se intenta involucrar en el medio a los alumnos ya que despierta la inquietud de aprender más. Este proyecto como se mencionó anteriormente intenta quitar el estigma de que la programación es algo muy complicado o incluso mágico, intenta hacer que los niños comprendan cómo es que funciona una red social, algún software, incluso que vean que en algún momento ellos pueden llegar a crear algo similar.

CS Unplugged se basa principalmente en un enfoque constructivista: los alumnos reciben desafíos basados en unas cuantas reglas sencillas y en el proceso de resolver esos desafíos descubren poderosas ideas por cuenta propia. No solo es una forma más memorable de aprender, sino que además los empodera para darse cuenta de que estas son ideas que están a su alcance. Las actividades también son bastante cenestésicas, cuanto más grandes sean los materiales, mejor.[21]

Dentro de esta página como profesor puedes acceder a una biblioteca de actividades diseñadas para distintas currículas que van desde música, matemáticas, español, ciencias naturales y muchos otros, algunas actividades pueden ser usadas para niños de 5 hasta 15 años.

2.3.2. Videjuegos en el Aula

Es imposible evitar voltear a ver los videojuegos en esta época ya que se han vuelto algo muy cotidiano, ya sea que se vea como una distracción o una oportunidad para obtener nuevas herramientas didácticas los videojuegos están presentes en el día a día, estos pueden ser unos excelentes recursos didácticos los cuales se han dejado de lado durante muchos años.

Los docentes han reaccionado de forma negativamente a la utilización de un material

considerado hasta hace poco como meramente lúdico en el desarrollo de sus clases, pero poco a poco este hecho va perdiendo adeptos y cada vez son más los profesores que quieren hacer un hueco entre sus técnicas metodológicas para los videojuegos. Ya que son la herramienta del presente y lo serán más aún en el futuro indicaba ya Morales Corral [22]. En el marco europeo del 2008 se formó una iniciativa llamada European SchoolNet que tiene como fin la utilización de videojuegos en las aulas, se publicó un manual entre todos los participantes donde Patrick Felicia [23] afirmaba que los videojuegos no son los sustitutos de las clases convencionales pero aportan valores y formas de educación que las clases tradicionales no pueden aportar de igual manera. Además los videojuegos cuentan con un componente motivacional muy importante que no se debe olvidar. Por tanto los videojuegos y las clases convencionales se complementan a la perfección tras observar los resultados de esta experiencia. [24]

Los videoJuegos pueden ser usados para el aprendizaje de distintas currículas como lo puede ser; Civilización III, que fue un videojuego lanzado en el 2002 por Infogrames para pc's con sistema operativo Windows, o Age of Empires III, lanzado en 2005 por Microsoft para la plataforma de Windows ambos tenían un contexto histórico, estaban situados en diferentes épocas y los alumnos tenían que desenvolverse dentro de estas de forma que poco a poco iban aprendiendo historia. Spore lanzado en el 2009 por Atari para multiplataformas es un videojuego que presenta la ciencia como algo muy simple que permite a los niños acercarse a esta y despertar en ellos el espíritu científico, como estos dos ejemplos se pueden encontrar muchos más en distintas áreas como lo puede ser ética y civismo, deportes, artes, etc.

Los videojuegos poco a poco son aceptados como una herramienta en el aula ya que sin perder su carácter lúdico y motivador permite al docente usarlos como un arma para transmitir algún conocimiento.[24]

El 16 de Octubre de 2012 se llevó a cabo un estudio en Madrid a cargo de aDeSe (Asociación Española de Distribuidores y Editores de Software de Entretenimiento) comprobaron la utilización de los videojuegos en la etapa de Educación primaria, viendo la influencia de los juegos en el comportamiento de los alumnos así como sus resultados académicos y el desarrollo de ciertas habilidades en las que se veían favorecidos por utilizar videojuegos. Se comprobó que más del 75 % de los profesores encuestados consideran que los videojuegos tienen una gran capacidad pedagógica y su introducción en el aula puede funcionar como un estimulante método de enseñanza, Pudo verse que el 33 % de los encuestados había usado videojuegos como herramientas didácticas y el 99 % pensaban repetir la experiencia. El 79 % del profesorado aprueban el uso de los videojuegos en el aula y lo consideran eficaz.

Opinan que los videojuegos favorecen el desarrollo de habilidades cognitivas 87,5 %, psicomotoras 80 % (destreza visual y discriminación perceptiva, coordinación espacial y lateralidad...), capacidades personales 64,2 % (autonomía, autocontrol, autoestima, creatividad...), sociales 49,9 % y habilidades y transmisión de valores 33,9 %.

En el caso de los padres, la encuesta pudo comprobar que un 20 % se oponía al uso de videojuegos en las aulas por motivos de falta de coordinación en los centros [24].

2.3.3. Videojuegos en el Aprendizaje de Programación Orientada a Objetos (OOP)

El uso de videojuegos como herramientas de apoyo metodológico y pedagógico son día a día un poco más comunes alrededor del mundo, distintas instituciones de educación superior las han ido implementando en sus enfoques pedagógicos, a continuación se exponen algunas:

- En la Escuela Colombiana de Ingeniería “Julio Garavito” se ha logrado experiencia en la enseñanza del paradigma orientado a objetos a través de juegos [25], donde los estudiantes desarrollan un modelo de clases que represente una pista de baile, compuesta de bailarines que pueden realizar una serie consecutiva de 7 pasos o “estilo” de baile, haciendo uso de una serie de métodos básicos definidos en la clase Bailarín, como subir y bajar tanto los brazos como piernas; el proyecto final es el juego de invasores, cuyo propósito es implementar un marco de trabajo (framework) con toda la lógica de un juego donde una nave controlada por el jugador se enfrenta a una diversidad de elementos, como los asteroides y elementos autónomos que se desplazan con determinados patrones y atacan al jugador en caso de detectarlo. [26]
- En la Facultad de Ingeniería de la Universidad ORT Uruguay [27], se presentan diversas líneas de trabajo en la Cátedra de Programación de la Facultad de Ingeniería, relacionadas con la enseñanza y aprendizaje de la programación. Estas líneas son el diseño de entornos de aprendizaje basados en la gestión del conocimiento, el uso de morfismos, el diseño y aplicación de “Kinesthetic learning activity” y el uso de Scratch en dos cursos de Ciencias de la Computación 1. Esta última herramienta se utiliza en las primeras semanas de los cursos con el propósito motivar y mejorar las experiencias iniciales con la programación, tratando de detectar su influencia en las puntuaciones y las tasas de deserción escolar en comparación con los cursos normales. Además, se desarrollaron guías detalladas de laboratorio, ejercicios, pruebas y formas de preguntas. Al contrastar los resultados con los cursos normales, se encontró que los estudiantes usuarios de Scratch denotan una motivación más alta, pero no hubo evidencia estadística de las tasas de abandono ni de las puntuaciones obtenidas.
- En la Universidad de Melbourne (Victoria, Australia) se tiene en cuenta que la mayoría de los estudiantes de Sistemas de Información de los primeros niveles no están motivados para la POO, por eso la idea de completar un juego de tablero 2D como materia de un semestre los animará en dicho aspecto [28]. Este trabajo fomenta la reutilización de componentes de software y permite a los estudiantes concentrarse en cuestiones de interfaz y diseño, sin dejar de satisfacer las necesidades de aprendizaje.

- En la Universidad de Corea (Seúl, Corea del Sur) se trabaja con el Juego de la Tarjeta Smalltalk (SCG)[29], que se basa en reglas de conversación de uso diario. Los participantes pueden entender los conceptos de la orientación a los objetos en forma evolutiva sin un conocimiento previo de la POO. En el transcurso del juego los participantes no utilizan términos técnicos como clase, herencia, instancia u otros asociados al paradigma, sin embargo usan escenarios que los conducen naturalmente al desarrollo de los conceptos por sí mismos y con sus propios términos.
- En la Universidad de Kettering (Michigan, USA), Baibak y Agrawal [30] sugieren diferentes clases de juegos para ser asignados en cursos de Ciencias de la Computación. Esto beneficia el plan de estudios porque se generan ambientes de aprendizaje divertidos y atractivos, brindando una base sólida para la comprensión y creación de algoritmos. Además, se sugieren diferentes tipos de juegos que podrían ser asignados en un curso, así como los tipos de algoritmos que pueden ser aprendidos en este proceso.
- En la Universidad de Roger Williams (Bristol, Inglaterra) y en la Escuela de Negocios Gabelli de la Universidad de Fordham (New York, USA)[31], en lugar de usar lenguajes tradicionales como Java o Visual Basic, se esta usando Alice, un ambiente de programación en 3D donde los objetos son humanos, animales u objetos de fantasía que habitan y actúan en mundos reales o imaginarios. Los estudiantes pueden utilizar Alice para la programación de animaciones o juegos interactivos. El método de Alice de "arrastrar y soltar" se ha reportado como una ventaja sobre la escritura de comandos.

2.4. Comparando Herramientas

Catacombs. Es un juego tridimensional, multijugador en el cual a través de los objetos de este los estudiantes aprenden cómo declarar variables, usar condicionales simples y anidados. De acuerdo a los escenarios del juego cada jugador representa un hechicero, dichos personajes necesita rescatar a dos niños los cuales estarán atrapados en las catacumbas. Con el fin de cumplir esta meta, los hechiceros deberán responder algunas preguntas de opción múltiple, para tratar de completar un código de programación el cual les ayudará a terminar su misión. Las respuestas de las preguntas automáticamente generan líneas de código ejecutable en micro-language Si la respuesta es correcta, los hechiceros pasarán de nivel, si no lo es, se les proporciona una explicación acerca del tema de la pregunta y se les dará otra oportunidad. El juego proporciona los puntajes de cada jugador y mensajes emergentes con explicaciones acerca de condicionales anidados [32][33][4].

Saving Princess Sera. Es un juego bidimensional el cual involucra a los estudiantes por medio de mensajes dirigidos al jugador. Cada jugador tendrá que intentar salvar a la princesa llamada Sera, quien fue secuestrada por un monstruo llamado Gargamel, en su cumpleaños 17. Los estudiantes deberán completar algunas tareas conforme la trama del juego avanza. Los jugadores irán completando líneas de código que resultará en un programa ejecutable el cual tendrán que organizar arrastrando y soltando las líneas de código que anteriormente habían completado. De esta manera los estudiantes aprenden de forma rápida de introducirles el algoritmo de ordenamiento de quick-sort [33][32][4].

EleMental: The Recurrence. Es un juego tridimensional, su objetivo principal es enseñar a los estudiantes como implementar recursión y el recorrido de arboles binarios DFS con el uso de lenguaje C#. El jugador tendrá que navegar por un árbol binario virtual empleando búsqueda por profundidad (DFS) y completar tres tareas en las cuales tendrán que aplicar recursividad. Durante el juego los estudiantes cuentan con la ayuda de dos avatares llamados Ele y Cera. Por ejemplo cuando el estudiante hace un código Ele recorre el árbol binario de acuerdo a como el código lo dice, mientras Cera explica exactamente qué parte del código es la que genera el movimiento de ese momento [34][4].

Wu's Castle. Es un juego de roles bidimensional cuyo objetivo es enseñar a los estudiantes el uso de ciclos o bucles y arreglos a través de actividades interactivas. Cada jugador es un hechicero que puede controlar un ejército de muñecos de nieve. Los jugadores deberán corregir errores lógicos en líneas de código escritas en lenguaje C++. El juego permite cambiar los parámetros de matrices dentro de ciclos y mover sentencias si se necesita dentro de los ciclos[35][4].

Robozzle. Es un rompecabezas en línea, proporciona una serie de comandos predefinidos los cuales están listos para ser usados y no muestra ninguna clase de código. De acuerdo al escenario del juego, los usuarios deberán desarrollar funciones que les ayuden a completar cada una de las tareas que deben cumplir en este mundo virtual. Los Usuarios pueden compilar sus funciones y observar cómo es que su personaje se mueve en medio del mundo, de esta manera les es mas facil detectar sus errores y corregirlos[36] [4].

LightBot. Es un rompecabezas en línea muy similar a Robozzle. Incluye series de comandos predefinidos, no genera código o utiliza algún lenguaje de programación en específico, Los usuarios tienen que completar tareas, por medio de funciones que van armando ellos, estas funciones constan en hacer que el personaje se mueva de un punto a otro del mundo y encienda una luz. Una vez que el usuario haya completado la tarea pasa al siguiente nivel, el cual requerirá hacer una función más compleja[37] [4].

TALENT. Su principal interés es enseñar la declaraciones de condicionales y ciclos en forma de algoritmos a través de un micro-language. Cada jugador es un arqueólogo que tiene que navegar dentro de un escenario para cumplir con ciertas tareas y recolectar objetos en locaciones específicas los cuales se exhibirán en el museo. Los estudiantes pueden arrastrar y soltar líneas de código en un editor según ellos lo crean conveniente. TALENT tiene un asistente el cual actúa como mentor y ayuda a los estudiantes con pistas según

ellos las necesiten para así pasar a la siguiente misión[38][4].

Robocode. Es un juego con entorno bidimensional, cuyo objetivo principal es enseñar a programar usando el lenguaje java. El juego cuenta con un editor, robots y una arena virtual, los estudiantes deberán programar a su robot para ganarle a otro que se encuentra en su misma arena. Mientras los estudiantes preparan a sus robots para el combate estos se familiarizan con los comandos básicos de estructuras de programación, y la programación orientada a objetos[39][4].

M.U.P.P.E.T.S. Está desarrollado en una página web colaborativa, con la cual se pretende hacer que los estudiantes se familiaricen con la programación orientada a objetos por medio del lenguaje Java. Los usuarios crean un robot que tendrá que pelear con otro robot dentro de una arena virtual, interactúan con los objetos que ellos crean además de escribir y compilar líneas de código por medio de un entorno de desarrollo integrado que incluye comando en la consola [40] [4].

Prog & Play. Es una librería que se encuentra en página web, es un juego de estrategia en tiempo real, que constantemente te hace interactuar con otros jugadores. Los estudiantes pueden programar sus propios personajes y formar alianzas con otros jugadores para prevalecer en el juego. Prog & PPlay permite a los estudiantes escoger el lenguaje de programación con el que quieren trabajar, cuentan con Ada, C, Java, OCaml, Scratch y Compalgo[41][4].

PlayLogo3D. Es un juego de rol tridimensional que permite la interacción entre múltiples usuarios con el fin de enseñar los conceptos básicos de la programación. Los usuarios deben programar los movimientos de sus personajes en líneas de código con el lenguaje de programación LOGO, para así poder pasar al siguiente nivel. Más específico, el mundo virtual en el cual tienen que interactuar se encuentra en la galaxia de Andrómeda, y se llama X-15 [42][4].

Gidget. Es un juego que se encuentra en una página web, donde los estudiantes pueden programar usando un lenguaje de programación simplificado, que fue creado específicamente para el juego, conforme el juego pasa los usuarios aprenden cómo diseñar y analizar un algoritmo básico. Un robot llamado Gidget tiene problemas con una parte de su software y así no puede completar sus tareas. Por lo tanto, los estudiantes son llamados a ayudar a Gidget a arreglar las líneas de código que tienen error, o a completar las líneas que están con hoyos. Por medio de este proceso, los estudiantes reciben una retroalimentación constante del proceso[43][4].

2.4.1. Karel

Karel es una herramienta de emulación la cual por medio de analogías permite a los usuarios aprender a programar de forma estructurada, durante la experiencia del juego el usuario es un robot; tiene que desplazarse en un ambiente el cual puede variar de acuerdo a la dificultad de la actividad, ya sea desde salir de un laberinto o encontrar comida, etc.

Juego	Elementos de Programación	Lenguaje de Programación	Actividades de Programación	Características Especiales
Catacombs	Variables simples y bucles	Micro-lenguajes	Preguntas de opción múltiple y rellenar líneas de código.	tridimensional, multijugador, tabla de puntuaciones, mensajes explicativos del personaje.
Saving Sera	Condicionales, recursión	Micro-lenguajes	Rellenar líneas de código y ordenar código. Recorrido por profundidad y el movimiento de un árbol binario	Construcciones bidimensionales
EleMental	Recursión, recorrido por profundidad (DFS) algoritmo	C#	Completar ocho misiones	Construcciones Tri-dimensionales
Prog & Play	Programación estructurada, condicionales y bucles	Ada, C, Java, OCaml,Scratch, Compalgo	Manejo de arreglos, Manipulación de un personaje, identificación de errores logicos. Crear funciones por medio de movimientos de un personajes	Multijugador, infinitos escenarios
Wu's Castle	Bucles y arreglos	C++	Crear funciones por medio de movimientos de un personajes	Juego de Roll y interacción.
Robozzle	Funciones	No usa código	Crear funciones por medio de movimientos de un personajes	interactivo y Disponible
LightBot	Funciones	No usa código	Crear funciones por medio de movimientos de un personajes, arrastrar y soltar código	Desarrollado para sitio web, disponible

Juego	Elementos de Programación	Lenguaje de Programación	Actividades de Programación	Características Especiales
TALENT	Condicionales y Bucles	Micro-lenguaje	Escribir líneas de código en un editor	Mensajes con explicaciones
Robocode	Programación estructurada, Programación orientada a objetos	Java	Escribir líneas de código	Cada robot tiene su historia y se emplean distintas actividades en cada una, disponible Colaborativo, tridimensional,
M.U.P.P.E.T.S.	Objetos tridimensionales, Programación orientada a objetos	Java	interactúan con objetos de desarrollo, escribir líneas de código, compilar	panel de comandos, entorno de desarrollo integrado
PlayLogo 3D	Conceptos y comandos básicos de programación estructurada	Logo	Creación de avatar, Navega por la "gaxia" por medio de comandos	multijugador
Gidget	Análisis y diseño de algoritmos básicos	Micro-lenguaje	Solucionar errores lógicos de programas y problemas, interactuar con un robot	Desarrollado para un sitio web

Tabla 2.1: En esta tabla se muestra de forma compacta todas las herramientas a modo de comparación.[4]

Karel un software educativo libre, el cual está basado en los patrones pedagógicos para la educación virtual [44], de tal forma que cuando se plantea una actividad o problema es fundamental tener en cuenta que es lo que se desea desarrollar, si ampliar el conocimiento tecnológico o un aprendizaje más profundo en este caso sería fomentar un pensamiento abstracto y ordenado.

El programa Karel se dio a conocer en el año de 1981 con la publicación del libro Karel The Robot: A Gentle Introduction to the Art of Programming [8] escrito por Richard E. Pattis, su objetivo principal era poder desarrollar una herramienta que le permitiera involucrar a más personas en la programación estructurada principalmente el lenguaje pascal, posteriormente con ayuda de Jim Roberts y Mark Stehlik (ambos profesores en su momento de la Universidad Carnegie Mellon) ayudaron a realizar una segunda versión la cual integraba conceptos no solo de programación estructurada, incluye también descomposición de programación estructurada, esta versión salió en el año de 1995. Poco después Joseph Bergin (profesor de la universidad de Pace) retoma el trabajo de Pattis y junto a tres colaboradores más desarrollan Karel++ y Karel J ambos tienen cambios bastante notables ya que su enfoque es para introducir a los estudiantes a la programación orientada a objetos Karel ++ fue presentado con el libro Karel++: A gentle introduction to the art of Object-Oriented Programming [6] el cual es enfocado al lenguaje de programación C++ y Karel J se presentó con Karel J Robot: A gentle introduction to the art of Object-Oriented Programming in Java [45] enfocado al lenguaje java.

Karel vive en un mundo dentro de la computadora, este mundo es una cuadrícula de 100 renglones con 100 columnas. Dentro del mundo de Karel puede haber paredes y zumbadores. Los zumbadores son pequeños dispositivos que pueden estar en diversos lugares. En cada cuadro de la cuadrícula puede haber desde 0 hasta miles de zumbadores. Los zumbadores se representan como un número que indica la cantidad de dispositivos en cada cuadro. Dentro de este mundo se cuentan con 5 indicaciones:

- **avanza:** Al ejecutar esta instrucción Karel avanza un cuadro en la dirección hacia la que está mirando.
- **gira-izquierda:** Al ejecutar esta instrucción Karel girará un cuarto de vuelta hacia la izquierda.
- **apágate:** Esta instrucción le dice a Karel que debe apagarse y terminar la ejecución.
- **coge-zumbador:** Al ejecutar esta instrucción Karel recogerá un zumbador del suelo y lo guardará en su mochila.
- **deja-zumbador:** Al ejecutar esta instrucción Karel dejará uno de los zumbadores que tiene guardados en su mochila en el suelo.

2.4.2. Scratch

Scratch es software diseñado por un grupo de investigadores del MIT que lleva el nombre de Lifelong kinder garden, actualmente está disponible para todas las plataformas y en múltiples idiomas, está diseñado para personas de 8 a 16 años, posterior a scratch,

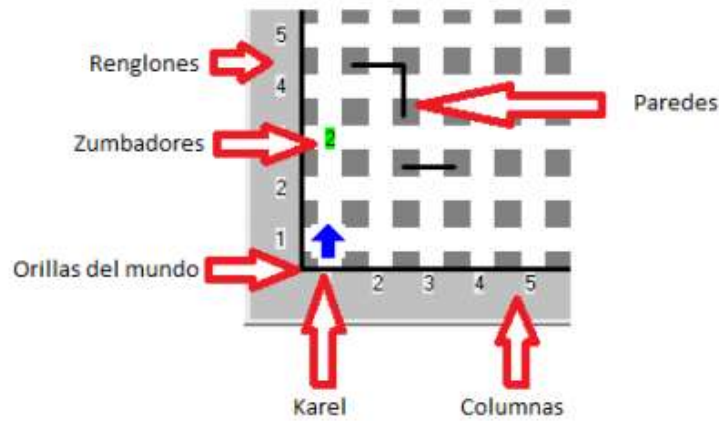


Figura 2.1: Interfaz de karel. Foto tomada de Tutorial de Karel el Robot Olimpiada Mexicana de Informática Abril 2015. César Cepeda.

crearon ScratchJr que es una versión diseñada para niños más pequeños de 5 a 7 años de edad, Scratch y Scratch Jr ofrece a los niños crear de manera interactiva sus propias historietas, juegos, animaciones y compartirlo con la comunidad, los usuarios desarrollan su creatividad, se enseñan a pensar, razonar de forma sistemática y a trabajar en equipo. Este software puede ser implementado en las asignaturas de artes, historia, matemáticas, computación.



Figura 2.2: Interfaz de Scratch. Foto tomada de [1].

2.4.3. Alice

Alice es un innovador entorno de programación 3D que hace que sea fácil crear una animación para contar una historia, un videojuego interactivo o un video para compartir en Internet. Es una herramienta de enseñanza de libre acceso, está pensada para facilitar

una primera toma de contacto de los alumnos a la programación orientada a objetos y permite a los estudiantes adquirir los conceptos fundamentales de programación. En Alice los objetos 3D (personas, animales, vehículos...) pueblan un mundo virtual y los estudiantes crean programas para animar estos objetos. En la interfaz de Alice los estudiantes arrastran y colocan elementos gráficos para crear un programa donde las instrucciones se corresponden con las sentencias estándar de lenguajes de programación orientada a objetos como pueden ser java, C++ y C#. Alice les permite ver, de manera inmediata, como se ejecutan sus animaciones programadas, y entender fácilmente la relación entre las sentencias de programación y el comportamiento de los objetos en sus animaciones. Mediante la manipulación de los objetos en su mundo virtual los estudiantes adquieren experiencia en nociones típicas de un curso de introducción a la programación.



Figura 2.3: Interfaz de Alice. Foto tomada de [2].

Capítulo 3

Diseño Centrado en el Usuario

El diseño centrado en el usuario (DCU), como su nombre indica, es una aproximación al diseño de productos y aplicaciones que sitúa al usuario en el centro de todo el proceso. Así, podemos entender el DCU como una filosofía cuya premisa es que, para garantizar el éxito de un producto, hay que tener en cuenta al usuario en todas las fases del diseño. Además, también podemos entender el DCU como una metodología de desarrollo: una forma de planificar los proyectos y un conjunto de métodos que se pueden utilizar en cada una de las principales fases. [46]

El término diseño centrado en el usuario nació en el laboratorio de investigación de Donald A. Norman en la Universidad de California San Diego (UCSD). Su amplia difusión se produjo después de la publicación del libro *User centered system design: new perspectives on human-computer interaction*. [47]

Aunque este enfoque puede ser usado en el desarrollo de cualquier tipo de producto, son los de índole tecnológico los que tiene una mayor importancia, ya sea hardware o software. El diseño centrado en el usuario DCU, como su nombre puede sugerir, es clave el papel que juega durante la todo el proceso de creación de un producto, dicho proceso puede ser dividido en tres grandes fases que más adelante es este mismo capítulo se explicarán.

El objetivo de hacer un diseño centrado en el usuario es poder proporcionarles a los potenciales usuarios un producto útil, funcional y que este hecho de a la medida sus necesidades particulares y estas puedan ser satisfechas de acuerdo a sus características. Por esto mismo es clave tener un proceso estructurado durante todo el trayecto de la creación de un nuevo producto, ya que obliga a tener siempre en consideración al usuario, planificar el modo de trabajo y saber que es lo que se debe de hacer en todo momento, es bueno resaltar que las fases que a continuación se mencionan todas son interactivas, es decir a pesar de que estas pueden estar definidas no es necesario verlas como estáticas o que deben de estar estructuradas de una manera en específica, esto dependerá del producto en particular y los objetivos del equipo de trabajo de esta forma permiten al equipo de trabajo moldear poco a poco el producto hasta obtener el objetivo deseado. [48]

La investigación y análisis de usuarios: Durante esta fase es necesario dedicarse a la recolección de de datos sobre los potenciales usuario, de esta manera se van estructurando todos los requisitos esenciales del producto, de acuerdo a las necesidades de los usuarios, cuanto más se conozca a al usuario mejor podrá llegar a ser el diseño del producto, por ello esta fase es de suma importancia en el desarrollo de un diseño centrado en el usuario, durante esta fase se pueden entrevistar a los posibles usuarios, realizar encuestas, focus group, entre otras cosas.[46]

El contexto de uso: Esta fase es muy importante, ya que varias de las metodologías usadas en el DSU están basadas esta fase, tienes que conocer el contexto de los potenciales usuarios, su entorno, esto es clave ya que determina muchas de las estrategias que se usarán en más adelante del proyecto, como el tipo de tecnología que uses, además que permite darle una mejor interpretación a los datos recaudados.[46]

Posterior a estas fases de investigación de usuarios tenemos las fases de **diseño y evaluación**, estas fases son puramente interactivas. Toda la investigación realizada en los usuarios, se traduce a crear perfiles, personas, tareas y/o escenarios, con ello comienza la parte inicial del diseño conceptual.[46]

Una vez que se tiene el diseño conceptual definido este se transforma en prototipos, los cuales deberán irse transformando de acuerdo a las pruebas que se realicen con los usuarios, al igual que el grado de fidelidad que estos nos puedan ofrecer, este proceso es cien por ciento interactivo y son usuarios reales los que realizan las pruebas, a este proceso se le conoce como pruebas de prototipos.

Debido a todo lo anterior se podría concluir que el DSU es una aproximación a empírica al desarrollo de productos interactivos, por ende es importante considerar que cada proyecto será completamente distinto y las tipografías, estrategias de trabajo, procesos y métodos de DSU que se emplean serán distintos e incluso se adaptarán a cada proyecto en particular, sin embargo siempre será necesario hacer un acercamiento e investigación de los usuarios potenciales y su contexto, ya que si no se conoce al usuario como se pretende poder diseñar algo que sea de utilidad para él.

El diseño centrado en el usuario es una filosofía y proceso que pone en primer plano al usuario y sus necesidades durante todo el proceso de diseño, haciendo de este un proceso interactivo, obteniendo retroalimentación a cada paso que se da de la mano de los usuarios y que tiene que ser usada para mejorar y adaptar el diseño de acuerdo a las necesidades y preferencias de los usuarios potenciales.

Como se mencionó con anterioridad el diseño centrado en el usuario es un conjunto de métodos y estrategias las cuales sitúan al usuario en el centro de todas las fases de diseño a en la Figura 3.1 se muestra a grandes rasgos los procesos que conforman el diseño, es importante entender que estos procesos no son absolutos, es decir tienen que ser modificados y adaptados de acuerdo a la necesidad, limitaciones y fin de cada proyecto, por lo general los factores que dictan el momento de empleo de dichas fases, son el presupuesto y el tiempo.

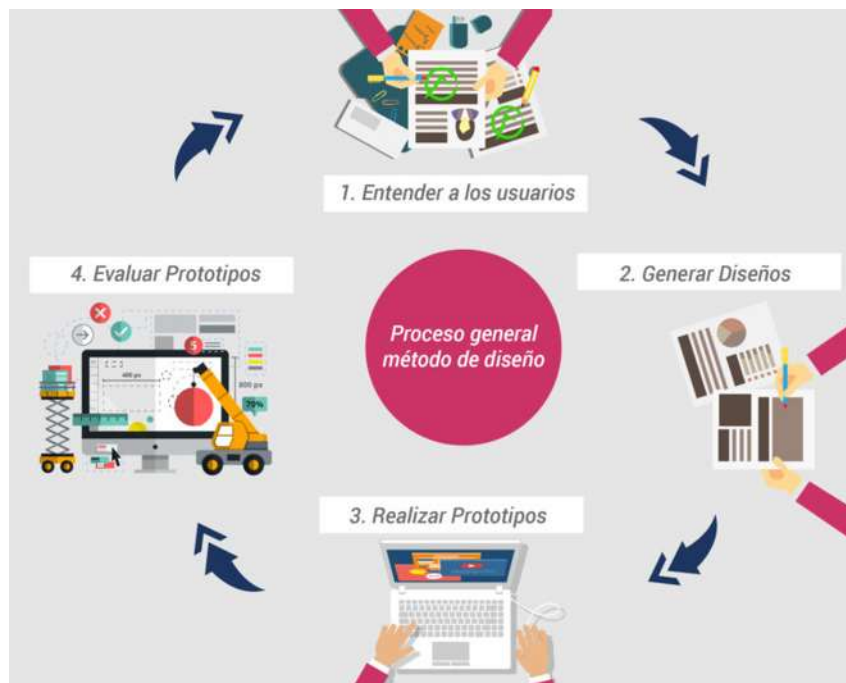


Figura 3.1: Descripción del Proceso de un Diseño Centrado en el Usuario. Fuente: Imagen realizada por TektonTech, iconografía freepik.es

3.1. Metodos de Investigación

Los métodos de indagación se llevan a cabo en las etapas de especificación del contexto de uso y de los requisitos. Se basan en involucrar a los usuarios en las diferentes actividades que se llevan a cabo y están orientados a la obtención de información para definir el producto o servicio. El conocimiento sobre los usuarios, sus contextos de uso, sus necesidades, objetivos y actitudes son imprescindibles para un diseño centrado en el usuario y para desarrollar aplicaciones y entornos usables [46].

Estos métodos pueden clasificarse por el tipo de información que permiten recoger: métodos cualitativos (observación, entrevistas, dinámicas de grupo) y métodos cuantitativos (encuestas y cuestionarios). Es muy importante tener claro que cada tipología permite obtener un tipo de información distinta y que debe ser analizada teniendo en cuenta sus ventajas y limitaciones.

A continuación se presentan algunas técnicas comúnmente utilizadas.

3.1.1. Observación e investigación contextual

Tal y como indica su nombre, la técnica observación investigación contextual consiste en la observación de los usuarios en su entorno habitual. Es una de las técnicas más útiles para conocer de manera objetiva qué hacen los usuarios, y en qué condiciones lo hacen. La información que se obtiene puede servir tanto para determinar sus comportamientos como para evaluar su usabilidad.

El trabajo de campo es de gran utilidad antes de empezar el desarrollo y en sus

primeras etapas. Este método permite conocer el comportamiento y la interacción de los usuarios con el sistema en sus contextos de uso habituales. Por otra parte, su principal inconveniente es que cualquier actividad de observación influye de un modo u otro en las acciones del usuario.

Hay una modalidad concreta de la observación denominada investigación contextual inquiry que se apoya fuertemente en el concepto de contexto de uso. Consiste en realizar la entrevista en el lugar donde se utilizará el producto interactivo y mientras se utiliza o se interacciona con él. Esta interacción del usuario proporciona la oportunidad de dirigir cuestiones de cariz más específico que permitirán entender sus dificultades y motivaciones. La investigación contextual es uno de los mejores métodos para descubrir y entender el contexto de los usuarios y cómo su entorno influye en sus interacciones.

La observación y la investigación contextual presentan el inconveniente de que hace falta desplazarse al lugar de interacción de los usuarios y que requiere una cantidad de tiempo considerable. Sin embargo, la información recogida con estos métodos es de una gran riqueza. [46]

3.1.2. Entrevistas

Las entrevistas también permiten obtener información de tipo cualitativo. Se llevan a cabo en muestras pequeñas, y suelen utilizar guiones poco estructurados y abiertos; no se busca representatividad, sino procesos y significado. No persiguen la cuantificación de las respuestas, sino una comprensión en profundidad sobre las necesidades, preferencias y experiencias de los usuarios con un producto o sistema.

Al basarse en la comunicación con el usuario, las entrevistas en profundidad permiten recoger todo tipo de información sobre comportamientos pasados, actitudes e intuiciones. No obstante, la información recogida puede estar afectada por diferentes sesgos e influencias. Por lo tanto, hace falta estar muy entrenado para llevar a cabo métodos como éste y poder obtener conocimiento riguroso.

Las entrevistas son más fáciles de realizar que la investigación contextual porque las personas son menos reticentes a desplazarse a un sitio neutro, pero entonces se pierde toda la gran cantidad de información que el contexto de uso nos permite recolectar. [46]

3.1.3. Encuestas

La encuesta es una técnica cuantitativa, a diferencia de los tres métodos anteriores. Por ello, se deben llevar a cabo entre una muestra representativa de usuarios. Para obtener los datos se usan formularios estructurados con preguntas que recogen idealmente todas las alternativas posibles de respuesta.

El diseño del cuestionario es clave para garantizar que la información obtenida sea relevante para los objetivos de la fase y del proyecto. Por ello, a menudo se opta por una fase cualitativa inicial cuyos resultados permitirán definir los ítems del cuestionario y las posibles respuestas.

Dada la importancia del cuestionario y sobre todo el hecho de que normalmente no hay contacto directo con los respondientes, es interesante hacer un pequeño test de la encuesta antes de enviarlo a toda la muestra. Con tres o cinco personas ya es suficiente para saber

si las preguntas se entienden, si las respuestas son pertinentes, el tiempo que se tarda en contestar el cuestionario, etc. Todo ello determinará la calidad de los resultados, reducir el abandono a media encuesta, etc.

Este método permite obtener información de manera remota, sin que los usuarios tengan que estar presentes. Presenta la ventaja de que la información que se obtiene tiene significación estadística. Por otra parte, a diferencia de los métodos cualitativos, en las encuestas es más difícil saber lo que los usuarios hacen realmente, la calidad y veracidad de la información que se recoge depende totalmente del diseño del cuestionario.[46] [49]

3.1.4. Focus group

Las dinámicas de grupo están formadas por las entrevistas que se llevan a cabo con un grupo de entre seis y ocho personas. La moderación de estos grupos es clave para recoger información de calidad y conseguir la participación de todos los asistentes. Igual que las entrevistas, el moderador o moderadora dispone de un guión para conducir la conversación en el orden que interesa y hacia los temas que se quieren investigar.

Las dinámicas de grupo suelen ser más largas que una entrevista y permiten recoger más diversidad de opiniones pero con menos profundidad. Sin embargo, también debe tenerse en cuenta que el efecto del grupo impide que cada participante exprese realmente lo que piensa, siente o cree. Esta información se obtiene mejor con una entrevista.[46] [50]

3.2. Indagacion al Diseño

Los perfiles de usuario, los personajes o personas y los escenarios son técnicas que nos acercan a los usuarios y a sus motivaciones, objetivos y situaciones de uso. Son técnicas que sirven para entender y analizar los usuarios y el uso que hacen de los sistemas interactivos y que se utilizan principalmente para orientar el diseño.

Para poder definir perfiles de usuario, personas y escenarios es necesario haber realizado previamente una recogida de requisitos y de análisis de los usuarios. Los resultados de entrevistas, observaciones, encuestas, dinámicas de grupo, etc., son analizados y adaptados para crear los modelos de usuario que se describen a continuación. [46]

3.2.1. Perfil de Usuario

Los perfiles de usuario son agrupaciones de usuarios según sus características. Normalmente, son el resultado de un estudio cuantitativo que ha permitido definir los porcentajes de cada perfil y agruparlos por elementos comunes. Estas características pueden basarse en aspectos sociodemográficos, actitudinales, de expectativas, etc. Las agrupaciones resultantes vienen marcadas por las preguntas de un cuestionario. Por ello, volvemos a recalcar la importancia de un buen diseño de encuesta y, si es posible, de partir de los resultados de un análisis cualitativo para definir las preguntas y respuestas del cuestionario de la forma más pertinente y adecuada.

Los perfiles de usuario incluyen a todos los usuarios. Posteriormente, y sobre todo con la creación de personas, los diseñadores escogen los perfiles más relevantes para el producto. [46]

3.2.2. Personas

Un personaje es la descripción de un usuario arquetípico que puede servir como guía en el proceso de diseño.

La técnica de los personajes se desarrolló originalmente como una técnica de ayuda al diseño, propuesta por Alan Cooper [46] en su aproximación al desarrollo de sistemas que tienen en cuenta al usuario, denominada goal-directed design. Un personaje se construye con precisión y rigor a partir de la información cuantitativa y cualitativa de la investigación de usuarios: segmentación, perfil de usuarios, observación, encuestas, entrevistas y dinámicas de grupo. Por lo tanto, primero hace falta investigar a los usuarios, analizar los datos obtenidos y, finalmente, modelar los usuarios en personajes. Se considera, de esta manera, que un personaje es un modelo de usuario.

El término original en inglés es “personaz se inspira en el método que utilizan los actores para desarrollar un personaje para así realizar una actuación más realista, de acuerdo con el papel que deben representar. Del mismo modo, los personajes se utilizan para que los diseñadores y los desarrolladores tengan en cuenta a los usuarios en las diferentes etapas del proceso de diseño y desarrollo y, por lo tanto, sirven de recordatorio de para quién se diseña y ayudan a tener en cuenta a los usuarios en todo el proceso de diseño y desarrollo. [46]

3.3. Diseño de prototipos y Evaluacion

Los perfiles de usuario, los personajes y los escenarios son formas de presentar los resultados obtenidos en el análisis de usuarios y la definición de sus requisitos. Esta información sirve para iniciar las actividades de diseño propiamente dichas.

3.3.1. Prototipado

El prototipado consiste en la construcción de uno o diferentes modelos del sistema o lugar web que se diseña. Estos modelos o maquetas simulan o ya tienen construidas partes del sistema final y se utilizan para llevar a cabo pruebas que no se podrían realizar hasta que el proyecto estuviera completamente finalizado. Estos modelos no pretenden reproducir el lugar web o el sistema en su totalidad, sino que normalmente corresponden a partes concretas que presentan una especial complejidad.

Normalmente, las técnicas de prototipado son llevadas a cabo por expertos en diseño centrado en el usuario con la colaboración del equipo de desarrollo. Las modalidades de esta técnica que se utilizan con más frecuencia son el prototipado de baja fidelidad, alta fidelidad, vertical y horizontal.

Los prototipos de baja fidelidad modelan elementos generales del sistema, sin llegar al detalle. Se pueden construir utilizando lápiz y papel y consisten en una representación esquemática del producto interactivo o del diseño de la interfaz. Así pues, no incluyen los aspectos de diseño gráfico de la interfaz ni los aspectos funcionales de la aplicación. Su objetivo es proporcionar una primera idea de cómo será la interfaz, de la disposición de sus elementos y de la visibilidad que deben tener. También se puede optar por pedir

que sean los propios usuarios quienes lleven a cabo los prototipos de baja fidelidad y así obtener información sobre sus preferencias y necesidades mediante el diseño participativo.

El prototipado de alta fidelidad trata de construir un modelo lo más próximo posible al sistema que se diseña y desarrolla. Este tipo de prototipo se utiliza para evaluar de manera más precisa aspectos funcionales y de usabilidad, tanto por parte de un experto en usabilidad como mediante un test con usuarios. Del mismo modo, se puede hablar de prototipado vertical y horizontal. El prototipado vertical consiste en la elaboración de un modelo de una parte concreta del producto o sitio web con la apariencia y funcionalidad que tendría si éste estuviera completamente finalizado. Eso permite llevar a cabo métodos de evaluación de la usabilidad sobre una parte específica o unas funciones determinadas. El prototipado horizontal se basa en la construcción de un modelo que abarca todas o casi todas las funcionalidades, espacios y menús del producto o sitio web, sin que estén implementadas y funcionen realmente. Sirve para evaluar no tanto el uso del sistema sino su alcance, la navegación y la arquitectura de la información. [46][51]

3.3.2. Técnica Mago de Oz

Esta es una técnica en la que el usuario de prueba interactúa con un sistema aparentemente inteligente, sin embargo quien toma el papel de dicho sistema y proporciona las respuestas es un operador humano al que se referencia como el asistente, la Fig.3.3 y Fig 3.2 es una imagen capturada durante una prueba de usuario en la cual se hizo uso de la técnica Mago de Oz, en este caso el usuario tenía que realizar un recorrido por el centro de la ciudad de Morelia, Michoacán, durante todo el recorrido el usuario era guiado y acompañado de su asistente inteligente “Elena” es en esta parte donde se hizo uso de la técnica ya que la chica de blusa naranja actuó como asistente.



Figura 3.2: Prueba de usuario haciendo uso de la tecnica de Mago de Oz. Foto tomada de [3]

Este tipo de experimentos o técnicas a menudo son usadas en interfaces de lenguaje natural.

La principal razón de usar esta técnica es que al tener un operador humano, tienes mucha más flexibilidad y oportunidad de interpretar cada una de las órdenes que el usuario da, además de guiarlo de forma más simple por el camino que se desea que el usuario tome, para así poder observar mejor los puntos clave a probar. La razón principal de esta elección fue tener un operador humano más interpretativo y flexible que permita a los participantes realizar gestos de la manera que desean, en lugar de tener que cambiarlos para que se ajusten a las limitaciones del reconocedor. Otra razón para elegir este enfoque es que no te exige ningún requisito para un período de entrenamiento, que puede cambiar la forma en que el usuario se desempeña a largo plazo. [52] [53][54]



Figura 3.3: Prueba de usuario haciendo uso de la tecnica de Mago de Oz, foto tomada de [3]

3.3.3. Test de Usuario

Los métodos de evaluación de la usabilidad por comprobación o test permiten realizar experimentos con los diseños y llevar a cabo pruebas y comprobaciones sobre ellos. A estos métodos y a sus diferentes modalidades se les denomina tests con usuarios. En general, el objetivo de un test con usuarios es proporcionar información de carácter cualitativo, que no cuantitativo o estadístico. Así pues, normalmente hay que dar más importancia a la interpretación de los resultados que a la elaboración de métricas o estadísticas. Mediante el test con usuarios se comprueba si las interfaces y los procesos están bien diseñados y si los usuarios encuentran obstáculos y dificultades para lograr sus objetivos.

Los tests con usuarios pueden variar desde el sencillo, con una sola tarea y pocos usuarios, a los más complejos, en los que se evalúa un conjunto de tareas con un volumen importante de usuarios repartidos en diferentes perfiles. Además, los tests pueden ser muy diferentes según la tipología y modalidad utilizadas.

El protocolo del *thinking aloud* establece pedirle al usuario participante en el test que vaya diciendo en voz alta lo que piensa, y de esta forma el conductor del test obtiene más información que complementa la observación de la actividad del usuario. Hay otros protocolos en los que, en lugar de pedir que los usuarios se expresen en voz alta, el conductor del test les va formulando preguntas relacionadas con las tareas o la interfaz. Normalmente los tests con usuarios se llevan a cabo en una sala preparada para ello que se denomina laboratorio de usabilidad, donde se graba la pantalla en la que el usuario interacciona con la interfaz y, de forma simultánea, se registran con una cámara sus expresiones. De esta manera, expertos en usabilidad pueden analizar el test sin necesidad de estar presentes.

Una de las principales desventajas del test con usuarios es el costo, especialmente el relacionado con las personas (el tiempo y los desplazamientos de usuarios y de los expertos involucrados) y además el hecho de que frecuentemente ocurre que los usuarios cambian su comportamiento sólo por el hecho de estar en un laboratorio y realizar un test. Una técnica que evita esta situación es el test remoto, que permite realizar tests con usuarios a distancia. La ventaja del test remoto es que la captación de usuarios es más sencilla, el usuario no pierde el contexto de uso habitual y, por lo tanto, el test no se sesga en este sentido. El principal inconveniente es que requiere de una infraestructura tecnológica –tanto del moderador del test como de la casa del usuario– que puede ser compleja y que, dependiendo de cómo sea, no permite llevar a cabo algunos protocolos de test, como el pensamiento manifiesto.

La principal ventaja del test con usuarios es que permite evaluar la usabilidad de una interfaz de manera bastante fiable y permite descubrir dónde los usuarios encontrarán obstáculos o dificultades. [46][50]

3.4. Realidad Virtual

La Realidad Virtual (RV) es una tecnología computacional que replica un medio ambiente, real o imaginario, y que simula la presencia física del usuario y un medio ambiente de manera que le permite interactuar con este [55]. La RV supone un avance y está teniendo nuevas aplicaciones en todos los ámbitos, pero es necesario también evaluar su impacto sobre el individuo, con el fin de saber qué efectos tiene esta tecnología sobre los usuarios cuando se utiliza de forma intensiva o cuando se abusa de ella, sobre todo en las aplicaciones lúdicas actuales, cada vez más virtuales o tridimensionales. No sólo el sector de ocio está utilizando la RV, sino que son cada vez más numerosas las aplicaciones inmersivas dirigidas a rehabilitación, adquisición de habilidades, tratamientos clínicos, etc. Aunque es una tecnología que lleva en desarrollo desde hace varias décadas, la disponibilidad a nivel consumidor de soluciones que implementen exitosamente la experiencia de RV. Uno de los proyectos más representativos que se encuentra en el mercado en cuanto a realidad virtual se refiere es Oculus. Oculus permite ser programado con Unity, además de tener sus propias librerías integradas para habilitar la programación para dispositivos móviles y PCs. Por otro lado, Google ha desarrollado el proyecto de Google Cardboard (GC) como una alternativa de bajo costo para el uso de aplicaciones de realidad virtual utilizando un teléfono inteligente [56]. Cardboard se refiere, en inglés, al cartón de embalaje con el

que GC está hecho. GC es en sí una plataforma que incluye tanto la montura hecha en cartón y el software utilizado en teléfono inteligente (usando Android o iOS). En cuanto a la montura, los usuarios de GC pueden crear su propia versión utilizando componentes de bajo costo (las especificaciones han sido publicadas por Google), o comprar uno hecho. El visor se usa colocando un teléfono inteligente dentro de este y mirando a través de los lentes de la montura como se muestra en la Figura 3.4.



Figura 3.4: Google Cardboard

Capítulo 4

Descripción del Experimento.

4.1. Observacion e investigacion contextual

La mayoría de los estudiantes del nivel medio superior en México (personas de entre 15 a 18 años) comienzan a programar, por primera vez en sus vidas.

Durante la edad de 15 a 18 años es muy habitual que los estudiantes interesados en el área de la computación tengan su primer acercamiento a la programación ya sea por iniciativa propia o por alguna materia si es que tienen la fortuna de contar con la orientación desde el bachillerato, sin embargo, muchos de los que realizan este acercamiento de manera autodidacta lo detiene ya que en su mayoría suelen frustrarse al no comprender algo bien o no encontrar una herramienta que logre captar su interés, por otra parte los que lo hacen de forma escolarizada suelen sufrir incluso más si no cuentan con un buen profesor, al percibir este conflicto fue que nació la idea de este trabajo.

4.1.1. Entrevistas

Una vez que la idea era clara: hacer algo que pudiera ayudar a los adolescentes en su acercamiento a la programación, fue necesario hacer una investigación de campo para saber las condiciones en las que se encontraban los posibles usuarios, para ello se contó con el apoyo del centro de bachillerato tecnológico y de servicios número 94 (CBTis 94) ubicado en el municipio de Pátzcuaro Michoacán, se seleccionó este lugar ya que cuenta con la especialidad de técnico en programación, cursando materias de esta rama desde el segundo semestre.

Para esta investigación de campo se realizaron entrevistas semi estructuradas con dos profesores que son los encargados de impartir la mayoría de estas materias de programación la primera fue la profesora Julieth Fuentes la cual comentó que ella solía tener un plan de trabajo “poco actualizado” hasta hace un año aproximadamente en el que ella cambió casi por completo su manera de introducir nuevos conceptos a los alumnos, ahora ella se enfoca en desarrollar primero un poco de la lógica computacional y los algoritmos, ella nos menciona “ Primero yo trato de que entiendan el concepto desde que es un algoritmo, para que puedan entender que la programación siempre se va a basar desde un algoritmo” la profesora comenta que le ha funcionado manejar pseudocódigos y diagramas de flujo, para que después ellos puedan hacer pruebas de escritorio, también nos dice que se suele

apoyar en distintos programas que te permiten leer y correr pseudocódigos.

Ambos profesores mencionan que se apoyan en el programa PSeInt la profesora también nos comentó “esa es la táctica ahora, enseñar desde el concepto de algoritmo, lo pasas por un software que te muestre paso a paso la salida para que ellos puedan ver que es lo que está pasando y puedan entenderlo”, ella menciona que de esta manera desarrollan un poco más su lógica computacional para una vez que esta parte está un poco más trabajada poder introducir un lenguaje de programación formal, ambos profesores coinciden que el lenguaje que mejor les funciona y creen más apto para comenzar es C ya que es permite tener un acercamiento a la programación estructurada.

El profesor Gustavo nos comentaba “el hecho del lenguaje C le permite conocer un poco más de las sintaxis y de las estructuras. . . en mi caso particular, necesito que aprendan programación estructurada y de ahí llevarlos a programación orientada a objetos” esto es algo en lo que nuevamente coinciden los profesores, ambos ven c hasta el momento en el que llegan a programación orientada a objetos ahí es cuando se trasladan a java, a diferencia de la profesora Julieth el profesor Gustavo menciona que a él en muchas ocasiones le facilita hacer actividades físicas en las cuales los alumnos logren entender que es lo que le piden a un algoritmo o a una computadora realizar, todo esto previo a programar o iniciar si quiera un pseudocódigo “son ejemplos de cosas que tienen en ese momento: como caminar, como resolver por ejemplo un ejercicio del area de una figura”.

Les preguntamos por las herramientas que utilizan actualmente en clases, el profesor dice que cuando los alumnos tienen problemas con aprender la lógica detrás de programar les motiva a utilizar scratch en su versión en línea, y que incluso lo usan en el salón de clases, utilizan raptor, para diagramas de flujo, pero que en los últimos semestres la cambia ya que la simbología de raptor no es la estándar. Al igual que también usa PSeInt, en cambio la profesora nos mencionó que ella usa estas herramientas y algunas otras “netsupport school”, este nos es muy eficiente, porque yo aquí puedo conectarme a la computadora de cada alumno y pues es más fácil que cuando tienen dudas que atravez de aquí me lanzan un salvavidas que aquí me parece, ... , esta es una herramienta para mi imprescindible”. Ella menciona que no realiza exámenes ya que no cree que sean muy eficientes y que en lugar de esto hace pequeños cuestionarios a los alumnos a través de la aplicación Kahoot!

Los profesores mencionaron que uno de los principales conflictos que suelen enfrentar es que los alumnos, no acostumbran hacer pseudocódigos o análisis de los problemas, lo primero que hacen es comenzar a teclear código sin siquiera analizar qué estructura es la que usarán. Al igual que tratan de memorizar las palabras de un lenguaje en lugar de comprender cuál es la acción que tiene dicho comando, en conclusión como dijo el profesor Gustavo “Es desarrollar la lógica que hay detrás” cuando se les preguntó qué es lo que ellos suelen hacer para solucionar estos conflictos; por una parte el profesor cree que basta con practicar, variedad de ejercicios y actividades incluso físicas para que los alumnos logren desarrollar la lógica detrás, en cambio la profesora Julieth dice haber optado por prevenir esto antes de que suceda, ella prefiere tomar si es posible todo un semestre para desarrollar únicamente la lógica y cuando comienzan a trabajar con un lenguaje lo hace con programación estructurada.

En cuanto a los temas que ellos consideran indispensables ambos tomaron rutas distintas que llevan casi al mismo destino, por una parte la profesora dice que es necesario

iniciar con el concepto de algoritmo y metodología de la resolución de un problema para posterior a esto entrar a programación estructurada definiendo las estructuras básicas. El profesor cree que es fundamental iniciar con lógica y después pasar a programación básica por medio de pseudocódigos y diagramas de flujo.

4.2. Personas/ perfil de usuario

EJEMPLO DE PERSONAS 1



“Me encantaría poder aprender a hacer apps, así podría hacer unos platos inteligentes para mis perros que pueda manipular desde mi teléfono”

Inés de la Fuente Macías.

Edad: 15 años.

Residencia: Pátzcuaro, Michoacán.

Ocupación : estudiante de preparatoria.

Inés vive con sus papás ambos trabajan, ella es hija única, su papá es Jefe de Departamento en una Secretaría del Gobierno de Michoacán, y su mamá trabaja como profesora de matemáticas en la preparatoria a la que ella asiste. Su materia favorita es computación, cuando era niña su papá le mostró scratch y le encanto, pronto comenzó a crear animaciones, de historias que ella misma escribía y se las compartía a sus amigos de la escuela, a pesar de que disfrutaba mucho de Scratch este se ha vuelto un tanto infantil y soso para ella intento buscar alguna otra plataforma que le permita seguir con su aprendizaje en programación pero Alice le parece demasiado complicado y las herramientas que su papá suele usar son muy aburridas y difíciles, “me gustaría que hubiera algún software que me permitiera saber qué es lo que estoy creando como lo hacía scratch pero que sea algo más divertido”, “quisiera poder aprender a desarrollar apps como mi papá”. Inés aún no tiene cien por ciento claro que es lo que le gustaría estudiar sin embargo sabe que tiene que ser algo que tenga que ver con la programación y el mundo de la computación, por esto mismo elegirá el bachillerato en físico-matemáticas con especialidad en programación, “creo que por fin podré aprender a programar de forma más estructurada y me proporcionarán herramientas adecuadas para hacerlo”.

EJEMPLO DE PERSONAS 2



“realmente no entiendo para que me puede servir programar cuando no voy a estudiar más”

Mauricio Avilés Sánchez

-Edad: 16 años

-Residencia: Tzintzuntzán, Michoacán.

-Ocupación: Estudiante de preparatoria y trabaja con su abuelo

La mamá de Mauricio es madre soltera, trabaja en un súper en Pátzcuaro, su abuela es ama de casa y su abuelo tiene un taller de chuspata, en el cual hacen piezas para vender al turismo, desde salas hasta canastas o esculturas, Mauricio todas las tardes tiene que ir a trabajar con su abuelo. Mauricio es un fan de las redes sociales, hace 3 años vinieron sus primos de texas y le regalaron una laptop, por las noches pasa un buen rato en youtube e intercambiando memes y stickers con sus amigos, él desea terminar la preparatoria para dedicarse por completo al taller de su abuelo. Todos los fines de semana por la mañana está en las canchas de fútbol de la unidad deportiva de Tzintzuntzán, jugando con su equipo, espera que esta vez sí ganen su torneo. Actualmente se cuestiona para qué escogió programación como especialidad, le cuesta mucho trabajo entender bases de datos y no se diga programación “cuando escogí mi especialidad creía que todo iba a ser muy divertido pero no entiendo nada, esta bien raro todo” “me gustaría que hubiera algo que me enseñara a programar o me explicaran mejor por qué los profes van bien rapido siempre y no nos tienen paciencia”

EJEMPLO DE PERSONAS 3



**“yo no elegí a la docencia
la docencia me eligió a mí
así que quiero hacerlo lo mejor
que pueda”**

Juan Luis Díaz Colin

Edad: 50 años

Residencia: Pátzcuaro, Michoacán.

Ocupación : profesor tiempo completo.

Familia: vive con su esposa.

Juan Luis es profesor tiempo completo desde hace 25 años en la cbtis 123, estudió arquitectura en la unam en la ciudad de México, su papá era profesor fundador del cbtis 123, cuando Juan Luis terminó la licenciatura su padre le heredó la plaza, Juan Luis jamás espero terminar dando clases pero su papá lo convenció de que era una apuesta segura ya que si aceptaba ese empleo tendría una entrada segura, y por las tardes podría dedicarse a la arquitectura que era su prioridad, así fue como Juan Luis terminó dando clases de matemáticas y posteriormente al ser de los pocos en la escuela que sabían manipular una computadora terminó dando clases de programación. “al inicio era bastante interesante ya que los alumnos aprendían conmigo y todos estamos muy entusiasmados por aprender, ahora todos mis alumnos solo prestan atención a su teléfono y no se que hacer para llamar su atención”, “he intentado implementar nuevas técnicas o herramientas pero ninguna me ha resultado”. Juan Luis vive con su esposa Martha y su gatita “carlota” disfruta mucho de sus tardes juntos en su patio mientras ambos leen un buen libro y toman té.

EJEMPLO DE PERSONAS 4



“Es imprescindible estar en constante actualización y experimentando con nuevas herramientas, aún más si de aprendizaje con tecnología se trata ”

Valentina Flores López

Edad: 40 años

Residencia: Morelia Michoacán.

Ocupación : profesor tiempo completo.

Valeria es madre de 2 hijos, está casada, su esposo tiene una constructora, ella desde muy pequeña supo que quería ser profesora, por azares del destino terminó estudiando físico-matemáticas y en la licenciatura descubrió la programación en fortran era algo muy retador para ella y divertido continuó formandose de manera amateur en el área, terminando la licenciatura hizo una maestría en cinvestav en matemática educativa, tuvo un par de problemas para encontrar trabajo, pero unos 8 meses despues de terminar la maestría encontró una oferta en una preparatoria la cual era una muy buena oferta pero las horas eran pára las materias de informática, postuló y gracias a lo poco que vio en la licenciatura y que ella había aprendido de manera autodidacta se quedó con el puesto, “la sorpresa fue que era más satisfactorio que mis alumnos entendieran lo que hacían y se apasionaron que el yo misma comprender a la perfección un algoritmo”. En sus clases a implementado un sin fin de herramientas y actividades, ha usado Karel, Scratch, “en algún momento intente con Alice pero es muy complicado para los chicos”, “ actualmente notó que me es cada vez más complicado captar su atención son mucho más hiperactivos sin mencionar las distracciones de sus móviles” a valeria le ha resultado muy bien últimamente usar Cs-Unplugged.

una vez que el equipo crea que ya programó su respuesta los miembros tendrán cardboard los cuales usaran para ver el resultado de su algoritmo en primera persona, si se equivocan podrán observar cómo es que su personaje se mueve en medio del laberinto y si su algoritmo llega a tener algún error esto lo podrán notar porque van a chocar contra alguna pared muy seguramente o algo, de esta manera el alumno desarrollara un nuevo conocimiento de forma constructivista, además de que en todo momento en una esquina del cardboard podrá ver que parte del código se está ejecutando, así cuando detectas un error en el algoritmo puedes regresar de inmediato a corregirlo, pero se enfrentarán con una penalización, cada vez que regreses.

Una vez que su tiempo se agote y/o logren salir del laberinto se premiarán los equipos y habrá una tabla de records.

La otra versión para juego individual, la cual está pensada para que sea usada por los alumnos que desean aprender a programar por cuenta propia y para los que desean tener un poco más de práctica en casa, en esta versión cuentan con distintos niveles y mundos los cuales va incrementando su dificultad, ya que se van integrando nuevos ítem (variables) y tanto en programación ya que será necesario razonar un poco más para resolverlo. Al jugador una vez que selecciona o asigna un nivel, se le presenta un laberinto visto desde la parte superior el tendrá alrededor de 2 a 10 min para resolverlo de acuerdo a la dificultad de este, la manera de resolverlo será muy parecida a la de Karel durante el primer nivel ya que solo se contará con dos indicaciones durante los primeros niveles sólo podrán avanzar al frente y a la derecha, además podrán hacer “funciones” ya que los comandos serán arrastrados en forma de bloques y estos podrán agruparse para crear tus propios bloques. En el segundo mundo se comienzan a introducir los tipos de variables, ahora el jugador programara un laberinto y resolverlo.

4.4. Prototipo de Baja Fidelidad

El día 07 de junio del 2019 se realizaron pruebas con usuarios para evaluar el prototipo baja fidelidad, de la versión para usar de forma individual descrita en la sección anterior, con alumnos del Centro de Bachillerato Tecnológico Industrial y de Servicios num. 94 (CBTIS 94), esta escuela se encuentra ubicada en la ciudad de Pátzcuaro, Michoacán. Este centro cuenta con alrededor de 1500 alumnos, contando con 6 especialidades técnicas y tres bachilleratos.

Para esta prueba participaron 4 alumnos de cuarto semestre de la especialidad de programación. Cada alumno fue pasando de forma individual a una de las aulas de la institución, la cual habíamos habilitado para la prueba, al inicio de cada prueba les especificamos que se trataba de una prueba de usabilidad (Fig.4.2), se les recalcó que lo que se estaba



Figura 4.2: Explicación.

probando en ese momento era la idea y el diseño no su capacidad o a ellos en cualquier aspecto, pedimos sus autorización para fotografiar y grabar parte de las pruebas, cada una tuvo una duración aproximada de 15 min, la prueba consistía en resolver un laberinto (Fig.4.3). Cada usuario fue colocado en el centro de un laberinto, dicho laberinto fue construido con el mueble que ya se encontraba en el aula, se colocaron las mesas en la forma que se necesitaba de acuerdo a cada laberinto y se usaron cartulinas para poder dar lograr más la sensación de realmente estar en un laberinto, una vez que el usuario se encontraba en el centro y se le había explicado qué era lo que se haría, se les proporcionó un iPad de papel (Fig.4.4) en la cual podían ver el laberinto con una perspectiva “aérea” contaban con post it con las indicaciones al igual que un área para poner estas y compilar (Fig.4.5), cada uno solucionaba su laberinto con las dos órdenes que contaban (girar a la derecha 90 grados y avanzar un cuadro), cada que ellos presionaban compilar les proporcionamos un cardboard que previamente había sido modificado para limitar la visión del usuario (Fig.4.6), con ayuda de una silla de oficina, se movía al usuario por el laberinto únicamente ejecutando las indicaciones que él había “programado”, (Fig.4.7) podían hacer corridas parciales si no estaban seguros, por cuestiones de tiempo permi-



Figura 4.3: primer prototipo papel.

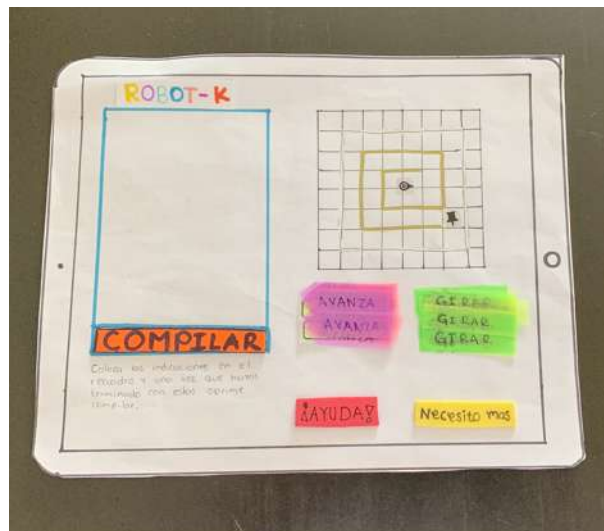


Figura 4.4: primer prototipo papel.

timos únicamente 3 por persona, una vez que terminaban sus tres compilaciones, se les hacían preguntas acerca de qué opinaban del proyecto y su experiencia. (Fig.4.8) Durante estas entrevistas se les preguntó acerca de qué opinaban sobre el proyecto y la experiencia que acababan de vivir, los usuarios comentaron que les parecía una buena idea, sin embargo existían cosas a mejorar, como por ejemplo el área de compilación era muy pequeña para las órdenes que tenían que colocar, alguno sugirió que además de tener la opción de realidad virtual se tuviera la versión “estándar” donde el personaje se desplaza por el laberinto solo en la pantalla. Se les cuestionó sobre si usaban asistentes virtuales ya que se pretende implementar uno más a delante.

4.4.1. Conclusión de la prueba

Con esta prueba se logró comprobar la eficiencia de la idea, el hecho de que el usuario use un visor de realidad virtual, tiene como consecuencia que este experimenta una

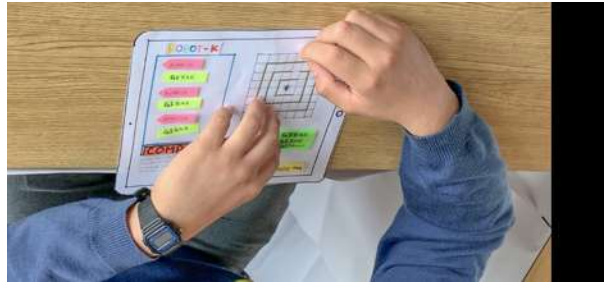


Figura 4.5: usuario colocando sus indicaciones.

inmersión, que provoca que tenga mayor concentración en lo que está realizando en ese momento, además de que los usuarios por cuenta propia tuvieron la iniciativa de que una vez que sufrían un “accidente” es decir su algoritmo era incorrecto para salir del laberinto, podían distinguir hasta qué punto su algoritmo estaba en lo correcto y tenía que ser corregido, añadiendo que les era mucho más simple razonar que era lo que había fallado ya que habían “vivido” ellos, justo lo que estaba sucediendo dentro de su algoritmo. Pudimos observar que los usuarios realmente razonaban dos veces que era lo que estaba sucediendo y lo hacían de manera “interactiva” ya que la mayoría de ellos si añadía una orden a su algoritmo, lo “probaban” con sus dedos recorriendo el laberinto como ellos creían que iba su algoritmo, de esta manera se daban seguridad de que estaban haciéndolo bien, después que hacían esto era que compilaban y si encontraban un error regresaban y volvían a hacer el mismo procedimiento solo que con las observaciones que se presentaron anteriormente, así que esto indica que la vista “superior” del laberinto es de suma importancia no solo porque les expone el problema si no también es un gran material de apoyo. Un usuario propuso que en tiempo real el personaje se desplazará en el laberinto mientras tú colocas órdenes, después pidió quedarse a observar la prueba siguiente y se acercó para decir que no le parecía adecuada su observación anterior ya que de esa forma perdería todo el razonamiento lógico, que sería mucho más fácil, que incluso lo podrían resolver solo adivinando. Lo cual indica que en efecto los usuarios pueden notar que están obligados a razonar que es lo que está sucediendo, es decir nuestro objetivo de desarrollar un pensamiento lógico y ordenado se está logrando.

4.4.2. Consideraciones a futuro.

Al realizar la prueba se lograron detectar algunas fallas y cosas en las cuales se puede mejorar, de tal forma que la experiencia del usuario sea mejor.

- La silla usada durante la prueba fue una silla de oficina con ruedas, la cual funcionó muy bien, sin embargo, sería mas cómodo para el usuario tener un apoyo para sus pies, ya que el mantenerlos en el aire implica cierto esfuerzo abdominal.
- El espacio que se dejó entre cada una de nuestras barreras que formaban el laberinto fue de un metro aproximadamente, en algunas ocasiones, mientras se giraba al usuario llegaba a chocar con las barreras lo cual hacía que se distrajera un poco de la experiencia y reducía el efecto inmersivo por ello, se recomienda dejar al menos un metro y medio entre cada “pasillo” del laberinto.



Figura 4.6: prueba.

- Durante la prueba para las barreras se emplearon las mesas del aula, para lograr un efecto de “paredes”, se colocó papel en todo el rededor del laberinto, al finalizar la prueba se pudo detectar que es irrelevante el uso de éste, la prueba se puede llevar a cabo sin problema alguno únicamente con las mesas o mueble de un aula.
- Para cada prueba se había asignado un estimado de 10-15 min, lo cual se pudo detectar desde el primer usuario que no sería posible ya que era muy “complejo” el laberinto del cual tenían que salir, con esto se pudo concluir que al menos los primeros ejercicios que realizan tienen que ser bastante simples y conforme los usuarios se van adaptando al juego es que la complejidad tiene que subir.
- Al inicio de las pruebas, se le permitió al primer usuario realizar todas las compilaciones que deseara, conforme los otros usuarios fueron pasando se detectó que no era lo mas óptimo y que se podía lograr que razonaran y analizarán más si solo tenían un límite de “intentos” para salir del laberinto así que se redujo únicamente a 3 compilaciones por jugador.



Figura 4.7: primer ejecucion.



Figura 4.8: entrevista a usuario.

Capítulo 5

Conclusiones y Trabajo Futuro

Gracias a los pruebas de usabilidad realizadas se puede concluir que, la idea realmente puede funcionar y ser aplicada en un salón de clases o en casa por medio de un smartphone, los profesores y alumnos entrevistados dijeron que realmente les gustaría hacer usos de ella, además de un par de profesores que mencionaron necesitar un software de este tipo, ya que les ayudaría a explicar mejor los temas básicos. El hacer uso de la realidad virtual, logra que los alumnos tengan una inmersión total lo cual ayuda que su atención esté completamente en el juego, de esta manera su concentración aumenta, al tener la sensación en primera persona durante la ejecución de sus comandos es más probable que los usuarios tengan un aprendizaje significativo, además de que la retroalimentación que les proporciona la realidad virtual les es suficientemente atractiva como para sí hacer uso de este software según lo mencionado en las entrevistas posteriores a las pruebas de usuario, los usuarios expresaron sus satisfacción con la idea y llegaron a mencionar que se sentían bastante atraídos a emplear un software del estilo.

Los usuarios comentaron durante la entrevistas que haciendo uso del software les parecía mucho mas simple el ordenar ideas, por comentarios como estos y la forma en la que resolvían los problemas probados con cada usuario se puede concluir que el software apoya a desarrollar un pensamiento lógico y secuencial, mejorar la abstracción y claro ayuda a comprender los principios básicos de la programación estructurada.

Durante las pruebas de usabilidad se logró ver que el proyecto podría emplearse un poco más, diseñar actividades donde los profesores únicamente usen los elementos que tienen a su alcance, como bien podría ser el mueble de un salón de clases para plantear problemas en los cuales los alumnos tengan que diseñar un algoritmo para solucionarlo y después puedan ejecutarlo con ayuda de un profesor de manera muy similar a la que se hace con CS Unplugged.

Debido a las limitaciones de tiempo, no se pudieron llevar a acabo las pruebas de alta fidelidad, para estas es necesario desarrollar un versión beta del videojuego, para la cual se emplearía el código libre de Unity, ya que es bastante amigable y los resultados pueden llegar a ser muy buenos en cuanto a realidad virtual se refiere, de esta manera se construirían los escenarios que se presentarán a los usuarios, para hacer las pruebas de alta fidelidad.

Apéndice A

Anexo

A.1. Focous Group

Focous group alumnos de primer semestre, con especialidad en programación del Centro de Bachillerato Tecnológico num. 94 Patzcuaro Michoacán. 21 de noviembre del 2018

Ok, bueno la pregunta con la que queremos iniciar es: **¿Qué esperarían ustedes aprender en un curso de programación?**

R:Utilizar el C++ y Adobe Fire. Ok. Me repites la pregunta. ¿Qué esperarían aprender en un curso de programación? Pues a programar ¿no?, los sistemas o los códigos. Aprender códigos para aprender a programar pues en sí programas o aplicaciones. Ok.

¿Qué fue lo que los motivó a escoger esta especialidad?

R:Yo voy a estudiar Programación en Sistemas. Ok. Yo voy a estudiar Artillería Naval. Ok. Yo necesitaba el bachillerato de físico-matemático. ¿Por qué? ¿Qué quieres estudiar? Ingeniería en Mecatrónica o Biomédica. O genial. ¿Y tú? Necesitaba el bachillerato físico-matemático. ¿Lo ocupabas? ¿Para estudiar qué? Físico-Matemáticas. Eh! Rompela. Estas hablando con dos Físico-Matemáticos. Este,

¿Qué es lo que esperarías aprender en un curso de programación?

R:No podría ser la creación de aplicaciones y sistemas. Sí ok. ¿Y tú? También. ¿Y tú? (no hubo respuesta) ¿Alguna pregunta que quieras hacer Fer? No paso, todo bien.

¿Qué tan importante consideran que es el saber programar? Osea en su vida diaria. Debe ser muy importante para tener uno bien

R:organizado las cosas. Prácticamente ya ahorita los tiempos que estamos viviendo todo es tecnología digital tendrías que saber para poder... Para robarme los 300 millones de pesos que se robaron unos compas. Sí eso es cierto. ¿Alguien más? Pues si te dedicaste a una carrera que se necesite la programación es indispensable porque pues no puedes estar en algo que no conoces osea es importante saber y saber lo que estás haciendo y también nos ayuda para programar nuestras actividades o sea saliendonos del tema de programar

computadoras y todo ese rollo para programar actividades como la agenda diaria, lo que ya estamos haciendo, salir a la escuela, en nuestros tiempos libres. Organización. ¿Tú tienes algún comentario por el momento? La siguiente pregunta que les podría hacer

¿Qué nivel de dificultad le pondrían a programar o a aprender a programar?

R: Del 1 al 10. Un 9. Un 8. El punto y coma se me olvida.

¿Están programando con java?

R: No yo estaba programando en mi casa con c++. Es que comenzaste con alojó y desalojó de memoria cada 5 min está como muy cañón. ¿Alguién más? Un 8. Un 8.

¿Qué herramientas usan sus profesores para enseñarles a programar o todavía no empiezan?

R: Todavía no empezamos. Ok.

¿Qué herramienta les gustaría que usaran?

R: Yo utilizo CodeBlocks se me hace muy fácil pero me gustaría uno que ya te ayudara con la compilación del código. Yo de eso no sé mucho. Ok. Yo de eso no sé mucho. Yo lo estudié por el bachillerato. Pero alguna herramienta que ustedes les gustaría.

¿Qué herramientas fabricarían si pudieran? Es una lista de deseos. Algo que les sirva para programar, ustedes ¿qué desearían?

R: Prácticamente ahorita sólo se hacen páginas sobre la creación. Una lista de códigos, que dijera este código sirve para iniciar una aplicación. O ver san Google. Cuando programas que te va gratis es mejor ir a la documentación dependiendo de qué lenguaje estés usando la documentación del sistema. Por ejemplo si están programando en java Oracle. Si están programando en py pues python. ¿Alguna pregunta Fer? Este, sí.

¿Qué tan eficiente consideran que puede ser una aplicación dinámica para enseñarles a programar es decir que no solo estén sentados en la computadora sino ustedes moviéndose? Como Kinect. Exactamente. Más o meno así como Kinect y Pokemon go. ¿Qué opinarían de eso?

R: Sería más fácil de aprender y sería más interesante. Llama más la atención y no te aburres tanto que estando sentado. No sientes que se te quema. Muy bien. ¿Ustedes cómo implementarían eso? Con implementar me refiero a ¿Qué cosas les gustaría que tuviera? Más que nada tiene que tener la dirección GPS y ubicación. Pero,

¿Qué cosas de programación le meterías, qué cosas para que tu aprendas?

R: La sintaxis. No sé te podría dar como sugerencias de códigos. Muy bien ¿Alguien más quiere aportar algo? Muy bien a ver. Fer preguntas. No todo bien. ¿Cómo te gustaría que te enseñaran a programar? Buena pregunta. ¿Cómo les gustaría? Se me vienen puras cosas masivas a la cabeza. Más bien sería como el uso de páginas pero eso sería muy común. ¿A qué te refieres con el uso de páginas? No elegí este bachillerato así, sino que es lo que necesito para

la Artillería por los grados, altura y rotación de la tierra, temperatura. Mi hermano estaba usando una aplicación que se llama ... porque le tocó hacer una página para poder pasar el año en la secundaria.

Focous group alumnos de tercer semestre, con especialidad programación del Centro de Bachillerato Tecnológico num. 94 Patzcuaro Michoacan. 21 de noviembre del 2018

Lo que estamos aquí es por un focus group estas cosas se suelen hacer cuando se piensa diseñar un software o algo por el estilo para que nos de retroalimentación. Nosotros les planteamos un par de preguntas ustedes las responden. Tienen toda la libertad de decir lo que quieran, criticar a quien quieran es completamente anónimo las cosas no van a salir de nosotros y pueden hacerlo con toda la libertad del mundo. Y cualquier cosa que necesiten decir haganlo. ¿Quieres empezar? o empiezo o ¿cómo Fer?. Empieza, yo no tengo las preguntas a la mano.

¿Qué es lo que esperan aprender en un curso de programación?

R: Yo digo que empezando por lenguajes de programación más que otra cosa y como aprender métodos como por ejemplo nosotros que estamos aprendiendo en java, las funciones principales y también encontrarle la lógica a la programación por que hay veces que por ejemplo para hacer un programa que haga algo en específico se requieren hacer ciertos pasos por ejemplo recuerdo hace algún tiempo un ejercicio donde nos pusieron hacer un programa para multiplicación y división de fracciones. Aprender ese tipo de cosas como qué orden y qué pasos y encontrarle la lógica. Tú. Es casi lo mismo. Esperas lo mismo. Aquí del CBTIS yo quisiera aprender nadamas lo básico que es de la programación. O sea que como dice mi compañero ese pensamiento lógico para poder programar. Es casi lo mismo que dijeron los compañeros aprender los lenguajes de programación y ya. Yo aprender a desarrollar aplicaciones. Cier-to, aprender a desarrollar aplicaciones, páginas web cosas que sean realmente útiles para una profesión. Ok.

¿Qué tan importante consideran que es aprender a programar?

R: Muy importante. Es muy importante al menos en la carrera que estamos tomando ya que en eso se basa vayamos aprendiendo teniendo las bases desde el principio bien. Bien puestas. De hecho es fundamental, la programación no sirve sólo para ser un buen desarrollador puede funcionar para muchísimos otros trabajos y muchísimos otros campos exactamente porque aprendemos a utilizar todas las funciones de un programa no sólo nos limitamos con la interface o las funciones básicas que tiene. Incluso siendo desarrolladores mejorar aplicaciones y todo eso. Yo diría que depende a lo que te vas a dedicar. Por ejemplo, si quieres ser chef no vas a dedicarte a programar. Ok, también en el campo de cocina usas algoritmos, las recetas son algoritmos. En la vida cotidiana también. Día a día usamos algoritmos pero sí.

¿Alguien más? Ok. Alguna pregunta Fer. No.

¿Qué nivel de dificultad le pondrían ustedes al aprender a programar del 1 al 10?

R: Eso depende mucho también del profesor por que hay quien se le hace fácil algunas cosas hay quien se le hace fácil todo. Hay quien se le complica absolutamente todo. Por ejemplo a mi al inicio cuando empiezo un nuevo lenguaje

se me complica absolutamente todo. Voy encontrando la lógica. Supongo que eso es dependiendo de las personas pero yo personalmente a la programación le pondría un no sé un 6 de 10 en cuanto a la dificultad. Yo digo igual que al inicio como cualquier cosa se te hace complicado pero al menos a mi en lo personal cuando voy aprendiendo un nuevo lenguaje y voy esté viendo más o menos como resolver los problemas se me complica más que nada los pequeños errores de alguna letra o punto y coma pero es por despiste no porque no entiendo más o menos qué es lo que está haciendo pero normalmente son errores de que me falta un punto. Entonces yo también le pondría dificultad 6 de 10. Yo diría que depende la persona, si tienes un razonamiento lógico obviamente se te va a facilitar mucho estar programando, para aprender otro lenguaje depende también mucho de la persona por ejemplo pasar de c a java no es tan difícil es cambiar palabras por otras palabras raras y poner un punto después de otra palabra mágica. Depende también de que tu investigues y seas autodidacta y con eso básicamente. Entonces ¿cuánto le pones? Un 7 de 10. Un 6. Ok.

¿Ustedes como intentan desarrollar su habilidad o lógica o su razonamiento lógico?

R: Porque veo que muchos se refieren a ello de que es una habilidad y que muchas veces también puede ser una gran dificultad, entonces cómo se enfocan en desarrollarlo o cómo intentan sobrepasar eso. Yo creo que sería con la práctica ya que entre más hagas alguna función o entre más hagas alguna aplicación o cierto tipo de lenguaje lo vayas usando en ciertos proyectos vas aprendiendo más de ese lenguaje, entonces a mí sería con la práctica o el estudio. Lo mismo que dijo el compañero. Es la práctica y el aprendizaje de todas funciones porque hay veces que parece que dos funciones son casi iguales pero te das cuenta que para tu programa específico una puede mejor que otra así que a parte de la práctica es supongo como la memorización ahora si que para aprender todas las funciones y la experiencia por que puede que la experiencia te diga que es más eficiente usar ciertos métodos en lugar de otros. ¿Ustedes? Igual.

¿Qué herramientas? ¿Consideran que las herramientas que usan sus profesores en clase son las adecuadas?

R: De momento sí porque como nadamas estamos aprendiendo apenas lo básico solo necesitamos los compiladores que tenemos y si en efecto son buenos. Los únicos problemas que tenemos a veces es que debido al servidor que tenemos en la escuela a veces se cruzan las sesiones y a veces yo abro la sesión de este amigo y él abre la mía. Esas son cosas que poco a poco el servidor tiene que ir corrigiendo.

¿Les gustaría que implementaran nuevas herramientas? ¿O tienen alguna herramienta que desearían que implementaran sus profesores?

R: Es que todavía no hemos visto todo completo entonces no podemos decir que a le falto esto.

¿O alguna que ustedes conocen vaya?

R: Yo personalmente quisiera desarrollar páginas web pero sé que se hace así todavía no lo vemos pero sé que sí.

En algún punto lo verás **¿Cómo te gustaría que les enseñaran a programar?**

R: Creo que sería mejor que las clases fueran más como podría decirse personalizadas porque es difícil como enseñar todos al mismo ritmo en una clase tan grande porque somos 40 alumnos y lo dividen a la mitad pero igual son como 20 y tantos y la programación es algo que si te atorras en una parte tienes que resolverlo porque no vas a poder continuar en la mayoría de casos y hay personas o a veces yo también me he quedado así de que no nos queda clara una cosa o no sabemos como empezar cierto código y ya no sabemos lo siguiente y se nos hace complicado, así que serían mejor como de que incluso hasta con un programa que te fuera enseñando paso a paso o en donde te quedaste, que el profe solo te explicara y ya despues que si tuvieras una duda pudieras corregirla no sólo diciendote a es que te faltó esto o lo otro o te faltó esta línea de código sino que te enseñen a que esta línea de código te falta porque esta cumple esta función en tal cosa y que te explicaran el porqué te hace falta y porque es necesario. Yo digo lo mismo que es una clase donde te explique más a detalle cual es el problema por hay veces que te atorras en algo y solamente te mencionan que te falta pero no te dicen porque o que es lo que va a hacer lo que te falte. O te enseñan un código y no te dicen exactamente cómo funciona. Exactamente. Lleva esto y ya. O va en el código y mágicamente funciona. Eso que dijeron mis compañeros. Que nos muestren como un código y que nos desglosen y nos muestren para que funciona cada palabra. ¿Ustedes? Igual.

Tengo una Isa. A ok adelante Fer. **De lo que les han enseñado hasta actualmente ¿Qué es lo que no les ha gustado o qué es con lo que ustedes han visto que casi no han aprendido mucho o que se les dificulta más para aprender los conocimientos que les están tratando de transmitir?**

R: Yo se que no es del todo la respuesta a la pregunta pero creo que lo que más se dificulta es por ejemplo yo personalmente en una materia en la de base de datos yo personalmente considero que domino la materia y se crear bases de datos completamente pero como que los profesores buena la profesora por lo que más se interesa no es por que los alumnos aprendan sino más porque los alumnos realicen los trabajos porque me ha pasado que una vez por ejemplo entre a una hora diferente y a pesar de que hice los trabajos perfectamente no me los quiso revisar y en los exámenes en realidad lo que califica no es lo que uno conoce de bases de datos sino la velocidad que uno tiene para responder. Porque lo hace con Kahoot. Si lo hace con Kahoot y no me parece correcto porque por ejemplo hay muchísimos que sabemos perfectamente cómo se hacen las bases de datos y esa no es una forma. A parte es media confusa la aplicación porque los colorsitos y ves los colores y tienes que voltear arriba para ver la respuesta. Hay veces que no te da tiempo y aunque sepas completamente la respuesta, o sea no importa que aprendas sino realmente que hagas

las cosas como ella dice y sino pues no. ¿Alguien más? No pues básicamente eso que a los profesores de aquí de bases de datos más que nada, los dos que nos han tocado les importa más poner un número en la boleta que en realidad que aprendamos. No le importa que el alumno sepa ya bien dominar todo sino que importa que haga lo que él dice como él lo dice y al tiempo que él lo dice. Sí, ese es problema de evaluación. Por eso dije que a lo mejor no era respuesta. Está bien. Eso es muy subjetivo. Depende mucho de la persona que te califica. Entonces, se supone que hay como una teoría ética que permita que esas cosas se regularizaran pero mucha gente la desconoce sobre todo los profesores que son los que la deberían de dominar. Pero sí tienen toda la razón y está muy bien.

Si ustedes pudieran pedir en este momento o sea hay una varita mágica atrás de ustedes y ustedes quieren aprender a programar lo que sea que lo hayan visto o no lo hayan visto o que se puede llegar a ver en algún punto de su vida. Y puedan pedir cualquier herramienta así lo que sea, que si ustedes dicen quiero que haya un Pokemon go que me enseñe a programar se puede hacer. Entonces, ¿Ustedes que harían? ¿Qué herramientas les gustaría que se desarrollaran o usar?

R:A mí en lo personal una herramienta que me enseñara más que nada a programar páginas web o aplicaciones móviles. A mi me gustaría más bien una herramienta que me ayudara a guiarme para crear alguna aplicación o algo algún juego o algo. Algo más intuitivo por que la programación se hace muy complicada y hay veces como que si hay códigos podría decirse innecesarios que si los tienes que tener pero que escribirlos completamente a mano es como no. Hay muchas herramientas que hacen eso, por ejemplo los constructores de Java pero hay veces que tienes que tomar programas en los que tienes que decirle dame un número y luego escribir el número y hay veces que tienes que escribir como 10 o 15 de esas opciones y nadamas cambia una variable y estaría bien que hubiera una opción como de crea 10 espacios o que te pida 10 números.

Muy bien. ¿Alguna otra pregunta Fer? **Entonces quisieran una aplicación que les ayudara como herramienta. ¿Qué tan eficiente verían ustedes una aplicación en la que no estuvieran pegados a la computadora y sentados que fuera un poco más didáctica y que pudieran hacer movimientos por ejemplo descifrar un laberinto cosas así de ese estilo pero con la idea de que ustedes pero aprendieran objetos, ciclos y cosas de ese estilo? ¿Se les haría interesante?**

R: Sería una forma más didáctica de aprender. Una forma más interesante y que no aburriría y no sería tedioso sino que más bien estarías complementando dos cosas que sería como el entretenimiento y el aprendizaje a la vez. Creo que entonces sería todo. Muchísimas gracias han sido muy útiles. Antes de que se me vayan.

Focous group alumnos de Quinto semestre, con especialidad programación del Centro de Bachillerato Tecnológico num. 94 Patzcuaro Michoacan. 21 de noviembre del 2018

Bueno pues nuestra idea es crear un prototipo una aplicación, cómo le podemos llamar, diseñar una aplicación que nos sirva para enseñarles a programar a ustedes. La idea de este focus group es como que ustedes nos digan qué dificultades enfrentan al aprender a programar o qué dificultades presentaron en el momento en el que les estaban enseñando a programar y también que herramientas nos sugerirían para implementarlas en nuestra aplicación entonces la idea es grabar información de tal forma que nos ayuden ustedes a complementar más las ideas que ya nosotros tenemos. Básicamente es eso. Entonces no sé si de entrada ustedes tengan alguna inquietud o algo. Todo lo que ustedes nos dicen aquí es completamente confidencial no se va a enterar ningún maestro, no se va a enterar nadie de hecho pues ni siquiera son las personas que vamos a sacar ni sus nombre, nada. Entonces ustedes tranquilos pueden hablar abiertamente y no hay ningún problema. La pregunta con la que quisiéramos iniciar es

¿Ustedes qué esperarían de un curso de programación o qué es lo que esperan?

R:Que enseñen los tipos de lenguaje como c++, java y todas esas cosas. Ok, que te enseñen tipos de lenguaje, alguien más. Pues primero para aprender a programar tienes que desarrollar como cierta lógica matemática entonces no puedes llegar nadamas a programar sino que tienes que aprender como lleva un procedimiento no? Tienes que aprenderlo primero y no llegar nada más así a programar. ¿Alguien más? La sintaxis.

¿Qué tan importante consideran que es el saber programar?

R: Mucho. Sí. ¿Por qué? Para entender todo lo que haces por que todos los que estamos en internet es programación. Yo creo como que hoy en día ya se usa mucho más eso no? O sea es más común que antes. A parte porque se usa en la vida cotidiana ya no hay algo como que diga no pues se va a hacer acá sin internet o computadoras. Es más la tecnología. No nadamas te sirva para ser programador sino para hacer otras cosas no? Porque sería como para resolver problemas o cosas así. Te ayuda en otros ámbitos. Y luego como ahorita la tecnología está pegando mucho pues todos deberían aprender a programar o por lo menos saber cosas básicas o algo así. Muy bien.

¿Qué nivel de dificultad le pondrían a programar del 1 al 10?

R:Es depende de cada persona porque no todos son, como dice mi compañero, se necesita mucha matemática o la lógica. La lógica de saber cómo funcionaría el programa. Entonces,

¿Qué nivel le pondría cada uno de ustedes?

R: Yo le pondría como un 7 porque no es tan difícil pero tampoco es trivial. Yo le pondría un 5 en mi experiencia.

Este, ¿Fer alguna pregunta? **¿Qué fue lo que más se les dificultó al momento de empezar a programar? O sea ¿la forma en como les**

estructuraron la clase o la forma en como les explicaba las herramientas el maestro? ¿O qué parte no les convenció?

R: Yo siento que el maestro, siento que iba muy rápido, como si ya lo tuviéramos que saber pues. Si lo dominaran. Prácticamente fue el maestro porque si se iba muy rápido y parte luego les decíamos que teníamos dudas y ella nos decía: pues ustedes busquen en dónde tienen el error ustedes ya deben aprender y así. Entonces pues no fue tan simple. En dificultad sería como aprenderse la sintaxis porque primero te tienes que aprender los tipos de variables que existen y luego por ejemplo que es lo que quieres que haga tu programa. Si quieres que te regrese como números enteros, flotantes o también todo lo que vas haciendo porque si haces algo mal ya te salió mal todo. Entonces pues sí es un poco difícil eso. Yo también diría que la sintaxis es lo que te mata. Yo digo que sería más la logística porque para hacerlo tienes que buscarle cierta lógica al programa, a la aplicación, que es lo que más complicado se me hace. **¿Qué herramientas usaba tu profesor a la hora de enseñarte o que herramientas usa?** El control de las computadoras nadamas. ¿Nadamas? Proyector. Ya averigüe quién es su profesor.

¿Qué herramientas a ustedes les gustaría que implementara? O de qué tipo de herramientas. Hagan de cuenta que ustedes en este momento pueden desear cualquier cosa que se implemente y se va a poder implementar o sea ya sea que exista o no exista. Casi casi que se pueden tomar la pildorita azul y todo el conocimiento va a estar en sus manos, algo así.

R: Yo diría que lo de la sintaxis y eso, el no tener que estar buscando como línea por línea, que ya te lo marque que está mal esa línea. Si quieres un tip, cuando compilas si te dice la línea donde está mal si es error de sintaxis. Si pero yo diría más como más gráfico. O a veces cuando cambias un if por un else como que altera y no te lo marca ahí. Es que ese es un error lógico. Hay dos tipos de errores pero muy bien muy buena idea. Es que eso si lo podemos implementar entonces muchas gracias, ¿Algo más? ¿Cuál era la pregunta? O sea, tú ahorita puedes desear cualquier herramienta para aprender a programar, la que sea, exista o no exista, tú ¿qué quisieras? Sería algo así como desarrollar juegos que te vayan enseñando, así sería más fácil y vas jugando y vas aprendiendo. O sería que desarrollaran como por ejemplo un IDE que tuviera como en la pantalla de inicio plantillas y entonces tú ya sabes lo que vas a hacer y puedes usar una de esas plantillas y estar copiando cosas y podrías no tener tantos errores. Sería más fácil. En lugar de escribirlo todo manualmente. Muy buena idea, alguien más. Es una lista de deseos a santa aprovechen. Puede que sus hermanos los usen. ¿Alguien más? No. ¿Fer pregunta? No.

¿Qué tan eficientes consideran que son los métodos que usan sus profesores?

R: Poco eficientes porque si pones atención si pero si por alguna razón no entendiste tienes que estar investigando por tu cuenta y a veces no es como lo quiere el maestro. Con la profa era mucho de como ella lo quería solo debe llevar esto y esto nada extra. Si te salía otra cosa eso no eso está mal. Si llegabas

al mismo resultado pero con diferentes métodos también estaba mal. Revisaba el código y decía es que aquí pusiste esto diferente y yo dije que tenías que ponerlo así. Y aunque compilara decía que no. Fer pregunta.

¿Ustedes utilizarían una aplicación de tal forma que no estuvieran pegados a la computadora que fuera un poco más dinámica, que ustedes pudieran caminar?

R: Hagan de cuenta que es como el Kinect y el Pokemon go los combinaron y ahora les enseñan a programar. Exacto, utilizando cosas de ese estilo para aprender orientado a objetos, las variables y cosas de ese estilo. ¿Se les haría interesante aprender de esa forma? Sería interesante pero yo siento que como muchos son muy flojos no funcionaría. Es que algunos para ponernos a programar agarraba me preparaba algo de comer y para quedarme un rato ahí sentado. Como que es difícil imaginarte salir a programar. Como que no sería, para personas como yo flojas, no me gusta mucho salir. Podemos desarrollar dos versiones. Una para los que quieren salir y otra por si no quieres salir. O hay personas que quieren aprender pero en todo caso no pueden salir. Claro. Una que sea simple porque por ejemplo si viene así con muchas cosas muchas palabras. Yo siento que si fuera un poco más didáctico más como entretenido como que te centrarías más en lo que estás haciendo. ¿Alguna otra aportación? Que hagan una serie en netflix. Una serie de programación.

¿Cuáles son los métodos que consideran más eficientes de todos los profesores que han tenido de programación? o ¿Qué es lo que les ha funcionado más?

R:Nadamas para hacer páginas web. HTML. Yo me cambié en segundo semestre y si me llevo a dar Gustavo y era un buen profe por que en la plataforma que tenemos de aulas virtuales implementaron una parte donde podías hacer tu código y ahí el profe como que no facilitó mucho eso porque ya no solo era mostrarselo en el momento sino que como era en el buscador lo podíamos hacer en nuestras casas y se lo podíamos enviar y esa era una herramienta muy útil. ¿Alguien más? ¿Alguna cosa que digan bueno esto no estaba tan mal? De Juliet ¿Qué más les gustaba? Nada. Daba décimas por participar. Cuatro participaciones era un punto. Alguna cosa que ustedes quieran agregar.

¿Estarían dispuestos a descargar un juego a su teléfono?

R:si

Sería todo. Han sido muy útiles. Gracias.

A.2. Entrevistas

Entrevista a Julieth Fuentes Profesora del Centro de Bachillerato Tecnológico num. 94 Patzcuaro Michoacan.

21 de noviembre del 2018

¿Qué tácticas o métodos educativos emplea en el aula cuando introduce nuevos temas de programación?

R: Primero yo trato de que entiendan el concepto desde que es un algoritmo, para que puedan entender que la programación siempre se va a basar desde un algoritmo, ya sea pseudocódigo, diagrama de flujo o cualquier otra cosas pero que entiendan el concepto de cómo se resuelve un problema, de que realmente programar es resolver algún problema de manera computacional y lo principal es que desarrollen su lógica computacional antes de que aprendan un lenguaje de programación.

Anteriormente los programas estaban diseñados para empezar con un lenguaje y empezábamos aquí con un lenguaje y ahora ese a cambiado o al menos yo lo he cambiado un poco, en lugar de empezar con un nuevo lenguaje en todo un semestre comenzamos con pseudocódigo con algún programamita como PSeInt, o alguno que maneje pseudocódigo y te de chance de hacer las pruebas desde los diagramas de flujo hasta las salidas en escritorio, que son las pruebas que harías a mano las hacemos ya con el software, entonces esa es la táctica ahora, enseñar desde el concepto de algoritmo, lo pasas por un software que te muestre paso a paso la salida para que ellos puedan ver que es lo que está pasando y puedan entenderlo, y ya que tienen la lógica computacional un poco desarrollada ahora si ya les ponemos un lenguaje, y lo principal es que se den cuenta que es indistinto el lenguaje, mientras sepas programar no importa el lenguaje que utilices.

¿Que dificultades suele encontrar con los alumnos cuando les está enseñando a programar?

R: Bueno, la principal cuando ya están en un lenguaje de programación, es que nunca desarrollan el paso previo, que es un pseudocódigo o un diagrama de flujo y quieren llegar a programar directamente en un lenguaje, se dan de topes porque no saben qué estructura va a aplicar si va a ser un if, si es un for, por que tu sabes que eso es igual en todos los lenguajes de programación, lo que tienes que entender es la estructura, y de hecho es una buena idea o al menos yo lo considero así, empezar con programación estructurada, ya una vez que entienden programación estructurada, ahora si ya vas con programación orientada a objetos y lo básico, que entiendan qué estructuras son las que conforman, la base de la programación y ese es mi principal problema.

¿Usted lo suele atacar yendo primero por estructura?

R: Si y como te comentaba hace rato empezando alomejor no luego luego con

un lenguaje si no ya hasta después de un semestre si es posible

¿Que tan eficiente considera que es su método?

R: Pues me ha funcionado en... un año solo tengo un año implementando este método pero me a funcionado más que como trabajaba anteriormente, ahora los chavos, por ejemplo ahorita ya estamos en programación orientada objetos y ya les puedo dar un problema y ya saben que van a aplicar un if, un for, un switch... ya saben que estructura van a usar por que ya lo tiene desde el algoritmo, entonces sí me ha funcionado.

¿Que tema considera fundamentales que aprendan?

R: iniciamos con conceptos básicos después emmm... algoritmo, metodología de solución ,de problemas, y ya estando en la programación yo iniciaría con estructurada y empezaria con definirles las 3 estructuras básicas, secuenciales, selectivas, repetitivas

¿Que lenguaje considera que es el ideal para introducir a programar a un alumno?

R: C, para mi seria C y despues java o cualquier otro.

¿Qué herramientas suele usar o implementar en sus clases?

R: Bueno primero un programa que yo creo que si conoces, es netsupport school, este nos es muy eficiente, porque yo aquí puedo conectarme a la computadora de cada alumno y pues es mas facil que cuando tienen dudas que atraves de aqui me lanzan un salvavidas que aquí me aparece, la computadora 7 te pidió ayuda, esta es una herramienta para mi imprescindible, cañón y la pantalla, últimamente utilizo mucho una plataforma llamada kahoot! la uso para hacer test rapiditos en equipos, ya no suelo hacer exámenes si no proyectos, aplicables en la vida real, más que comprobar que saben un concepto es que lo apliquen, ahorita precisamente estamos conectando una base de datos con netbeans y asi quee vean que si,que lo que hacen en un programa funciona en otro y asi... y lo que les estoy calificando es que van a un lugar.., la biblioteca y no tiene un software que controle el préstamo de libros y entonces con lo que saben hacen un programa que controle eso, básicamente es eso: mi herramienta principal, el netsupport schoolel ide que utilizo para enseñarles programacion que es netbeans, manejador de base de datos, ahora estamos viendo mysql, kahoot que no se llamen examen y no se asusten, entonces ya nadamas les hago un jueguito y listo y este... thatquiks, es otra plataforma para hacer pequeños exámenes y utilizó las aulas virtuales, aquí tenemos este... sobre moodle, tenemos un servidor de nuestras propias aulas virtuales aqui y el servidor es aqui de nosotros, entonces es, pues administrado por nosotros mismos y no hay bronca, hay otro no recuerdo como se llama ¿¿schoologi?! algo así es una plataforma igual de aulas virtuales que nosotros hacemos aquí en nuestra propia red y eso es todo lo que utilizo.

¿Estaría dispuesta en sus clases usar un aplicación o software tipo el que le comentaba?

R: Si, me sería muy util, he escuchado que mis alumnos aprenden a programar jugando pero yo nunca lo he usado, ellos son de mira aqui hay un programa que me enseña a no se armar cosas o así pero programando, ellos a veces lo usan pero yo no lo uso,es más yo no te puedo dar ahor amismo el nombre de un programa asi por que no tengo la referencia.

-De hecho es un buen proyecto, se que existen pero no las he usado

-Si hay varias pero queremos darle un twist

-Es una manera que ellos no se dan cuenta que estaba programando, pero aprenden a programar.

-y aprenden la lógica detrás

- y aprenden la lógica detrás y ellos están jugando y ya cuando razonan dicen, a este aquí utilizo un if a este aquí utilizo un for, alomejor no saben que es un for pero dicen lo hizo muchas veces, entonces sí me parece una excelente idea.

Entrevista a Gustavo Profesor del Centro de Bachillerato Tecnológico num. 94 Patzcuaro Michoacan.

21 de noviembre del 2018

¿qué tácticas de enseñanza suele usar a la hora de enseñar a programar?

R: Ejemplos reales, palpables, generalmente son ejemplos de cosas que tienen en ese momento: como caminar, como resolver por ejemplo un ejercicio del área de una figura que es algo básico o cómo hacer una suma.

¿te ha resultado su método?

R: Sí, generalmente lo que ocurre es que te pone a pensar un poco más a detalle qué estás haciendo en realidad y no asumir las cosas.

¿qué dificultades encuentra a la hora de que un alumno intenta aprender a programar?

R: La mayor parte del problema que yo me encuentro es que quieren aprender las palabras de un lenguaje de programación, no es así, es más bien desarrollar una lógica que hay detrás.

¿cómo lo soluciona?

R: Práctica y diversidad de ejercicios no sólo matemáticos, sino ejercicios de otros tipos, dando incluso actividades físicas para que los alumnos logren entender que no se trata de que solo se sienten a escribir si no de razonar procesos y detallarlos.

¿qué temas considera fundamentales aprender cuando se intenta programar?

R: Lógica, necesitan cosas como la parte de tablas de verdad, hay un tema no recuerdo el nombre exacto que es como de toma de decisiones y también es básico, la parte de no asumir situaciones, ejemplo la suma de dos números, muchos dicen tómame el primero y luego lo sumo al segundo y ya está el resultado ¿cuando le preguntaste cuál era el primero? ¿cuando por el segundo? he tenido casos que preguntan por el resultado y entonces qué caso tiene, si de pronto muchas de las situaciones es que los alumnos dicen yo ya me lo sé, tu si la computadora no, acuerdate que la computadora es un niño pequeño que necesitas decirle paso por paso que va a hacer.

¿que lenguaje de programación considera que es el mejor para introducir alumnos al mundo de la programación?

R: Principalmente utilizo C por que alomejor no es el más flexible, como ahí está python que es maravilloso pero el hecho del lenguaje c le permite conocer un poco más de las sintaxis y de las estructuras, python ya es un poco más

avanzado y les va ir quitando cosas, en mi caso particular, necesito que aprendan programación estructurada y de ahí llevarlos a programación orientada a objetos y así, e ir creciendo la dificultad, en cambio si comenzara con python que ya tiene muchas cosas prefabricadas, las van a omitir y más adelante cuando lo necesiten no saben usarlo por que ya estaba hecho en python y nunca lo necesitaron.

¿qué herramientas usa en sus clases?

R: La versión en línea de scratch si los alumnos gustan, raptor, PSeInt, en último semestre estoy dejando de usar raptor porque no vienen figuras estándar en un diagrama de flujo, trae su propia simbología y solo funciona bajo windows o sea no lo puedo usar en mi maquina.

¿estaría dispuesto a implementar un programa como el plateado?

R: Si, viendo las características y dependiendo de las opciones que me brinde la verdad es que si es una herramienta muy útil y sobretodo si esta en linea o multiplataforma.

Bibliografía

- [1] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, November 2009.
- [2] Wanda Dann Dennis Cosgrove. Alice 3, how to guide. 05 2012.
- [3] Juan A. Casimiro O., Izamar Gutiérrez A., Alberto Herrera L., and Isabel López H. Anhelo: Sistema para la creación de una memoria colectiva en la ciudad. *Avances en Interacción Humano-Computadora*, 0(1):66–69, 2016.
- [4] Christos Malliarakis, Maya Satratzemi, and Stelios Xinogalos. *Educational Games for Teaching Computer Programming*, pages pp 87–98. 06 2014.
- [5] Instituto Federal de Telecomunicaciones. Informes estadísticos, 2019.
- [6] J. Bergin, M. Stehlik, J. Roberts, and R.E. Pattis. *Karel++: A Gentle Introduction to the Art of Object-Oriented Programming*. Wiley, 1996.
- [7] Mitchel Resnick. Sowing the seeds for a more creative society. *Learning and Leading with Technology*, 35(4):18–22, 2007.
- [8] Richard E. Pattis. *Karel the Robot: A Gentle Introduction to the Art of Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1981.
- [9] Sean Choi. How to teach programming to teenagers, 2018.
- [10] Ma. del Carmen Chamorro. *Didáctica de las Matemáticas para Educación Infantil*. Person Prentice Hall., Madrid, España, 1st edition, 2005.
- [11] J. Piaget. *Introduction à l'épistémologie génétique*. Number v. 1 in Bibliothèque de philosophie contemporaine. Logique et philosophie des sciences. Presses Universitaires de France, 1950.
- [12] Z.T. Hernandez and H.T. Martínez. *Planeación y Control*. Económico Administrativo. Grupo Editorial Patria, 2014.
- [13] C. Chamorro and M.C.C. Plaza. *El aprendizaje significativo en el area de las matemáticas*. Documentos Para la R Series. Alhambra, 1992.

-
- [14] Orozco C. Moreno, F. *Teoría de la instrucción vs. teoría del aprendizaje significativo: contraste entre J. Bruner y D. Ausubel*. 2009.
- [15] César Coll. Significado y sentido en el aprendizaje escolar: Reflexiones en torno al concepto de aprendizaje significativo. *Infancia y Aprendizaje*, pages 131–142, 01 1998.
- [16] Miller Galindo Suárez. *Efectos del Proceso de Aprender a Programar con "Scratch".en el Aprendizaje Significativo de las Matemáticas en los Estudiantes de Grado Quinto de Educación Básica Primaria*. Tesis de maestría, Instituto Tecnológico y de Estudios Superiores de Monterrey, Julio 2014.
- [17] Luz Manuel Santos Trigo. *La Educación Matemática, resolución de problemas, yel empleo de herramientas computacionales*. Centro de Investigación y de Estudios Avanzados, IPN, CDMX, Mexico, 2011.
- [18] Manuel Santos-Trigo. High school teachers' use of dynamic software to generate serendipitous mathematical relations. 01 2010.
- [19] L.M.S. Trigo. *La resolución de problemas matematicos / The Resolution of Mathematical Problems: Fundamentos Cognitivos / Cognitive Fundamentals*. Biblioteca de la Asociación Nacional de Profesores de Matemáticas. Editorial Trillas Sa De Cv, 2007.
- [20] Manuel Santos-Trigo and Matías Camacho-Machín. Towards the construction of a framework to deal with routine problems to foster mathematical inquiry. *PRIMUS*, 19(3):260–279, 2009.
- [21] Computer Science Education Research Group. Cs unplugged, 2000.
- [22] Enrique Morales. El uso de los videojuegos como recurso de aprendizaje en educación primaria y teoría de la comunicación. 01 2009.
- [23] Patrick Felicia. Digital games in schools : Handbook for teachers, 2009. Complements to the study How are digital games used in schools? (European Schoolnet).
- [24] Javier Pascual Gutiérrez. *El uso de Video Juegos como herramienta didactica*. Tesis de licenciatura, Universidad de la Rioja, Julio 2013.
- [25] Cadavid H. Desarrollo de juegos como base para la comprensión de temas fundamentales de la programación orientada a objetos. 01 2006.
- [26] Ricardo de Jesús Botero Tabares. *LA LÚDICA DE JUEGOS EN EL APRENDIZAJE DE LA PROGRAMACIÓN ORIENTADA A OBJETOS: UN PROTOTIPO EN C*. Tesis de licenciatura, Universidad EAFIT, Marzo 2012.
- [27] I. Friss. Scratch: Applications in computer science 1.38th asee/ieee frontiers in education conference, saratoga springs, ny. 01 2008.
-

-
- [28] Balbo S. Goschnick, S. Game-first programming for information systems students. . in proceedings of the second australasian 65 conference on interactive entertainment (ie 2005). creativity & cognition studios press, sydney, australia, australia, 71-74. 01 2005.
- [29] S. Kim. Smalltalk card game for learning object-oriented thinking in an evolutionary way. in companion to the 21st acm sigplan symposium on object-oriented programming systems, languages, and applications (oopsla '06). acm, new york, ny, usa, 683-684. 2006. 01 2006.
- [30] T. Baibak. Agrawal, r. programming games to learn algorithms. 2007 asee annual conference exposition. honolulu, hawaii, 2007. 01 207.
- [31] W.B. McKenzie. An alternative approach to introductory object oriented programming: a case study in programming with alice. proceedings of the ais sig-ed iaim 2007 conference, montreal, quebec, 2007. 01 2007.
- [32] Richter H. Chaffin A. Godwin A. Powell E. Ralph T. Matthews P. Jordan H. Barnes, T. The role of feedback in game2learn, chi 2007, p.1-5. 01 2007.
- [33] Chaffin A. Powell E. Lipford H. Barnes, T. Game2learn: Improving the motivation of cs1 students. proceedings of the 3rd international conference on game development in computer science education, p.1-5, miami, florida. 01 2008.
- [34] K. Hicks D. Barnes T. Chaffin, A. Doran. Experimental evaluation of teaching recursion in a video game. in proc. acm siggraph symposium on video games, p. 79-86. 01 2009.
- [35] Barnes . Eagle, M. Experimental evaluation of an educational game for improved learning in introductory computing. sigcse bull. 41, 1 (march 2009), p. 321-325. 01 2009.
- [36] C. Li, F.W.B. Watson. Game-based concept visualization for learning programming, proceedings of the third international acm workshop on multimedia technologies for distance learning, p. 37-42, december 01-01, 2011, scottsdale, arizona, usa. 01 2011.
- [37] S. Piteira, M. Haddad. Innovate in your program computer class: An approach based on a serious game. osdoc - open source and design of communication workshop. 01 2011.
- [38] Grigoriadou M. Maragos, K. Exploiting talent as a tool for teaching and learning, the international journal of learning, volume 18, issue 1, pp.431-440. 01 2011.
- [39] P. O'Kelly, J. Gibson. Robocode problem-based learning: A non-prescriptive approach to teaching programming. acm sigcse bulletin, 38, 3, p. 217-221. 01 2006.
- [40] Bierre K Parks D. Phelps, A. Muppets: multi-user programming pedagogy for enhancing traditional study, proceeding of the 4th conference on information technology education, october 2003, lafayette, indiana, usa, p. 100-105. 01 2003.
-

-
- [41] Torguet P. Viallet F. Jessel J.P. Muratet, M. .experimental feedback on progplay, a serious game for programming practice”, eurographics, p. 1-8. 01 2010.
- [42] Arapidis C. Mpimpitsos M. Paliokas, I. Playlogo 3d: A 3d interactive video game for early programming education. third international conference on games and virtual worlds for serious applications, p. 24-31. 01 2011.
- [43] A.J. Lee, M.J. Ko. Personifying programming tool feedback improves novice programmers’ learning, conference on international computing education research (icer), august 8-9, providence, rhode island, usa, p. 109-116. 01 2011.
- [44] José María R. Patronos pedagógicos en educación virtual. *RED. Revista de Educación a Distancia*, pages 1–16, 2009.
- [45] J. Bergin, M. Stehlik, and J. Roberts. *Karel J. Robot: A Gentle Introduction to the Art of Object-oriented Programming in Java*. Dream Songs Press, 2005.
- [46] Enric Mor Pera Muriel Garreta Domingo. Diseño centrado en el usuario. page 58, 01 2011.
- [47] Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002.
- [48] Kimmo Tarkkanen, Pekka Reijonen, Franck Tétard, and Ville Harkke. Back to user-centered usability testing. In Andreas Holzinger, Martina Ziefle, Martin Hitz, and Matjaž Debevc, editors, *Human Factors in Computing and Informatics*, pages 91–106, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [49] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. Chapter 5 - surveys. In Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser, editors, *Research Methods in Human Computer Interaction (Second Edition)*, pages 105 – 133. Morgan Kaufmann, Boston, second edition edition, 2017.
- [50] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. Chapter 8 - interviews and focus groups. In Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser, editors, *Research Methods in Human Computer Interaction (Second Edition)*, pages 187 – 228. Morgan Kaufmann, Boston, second edition edition, 2017.
- [51] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. Chapter 10 - usability testing. In Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser, editors, *Research Methods in Human Computer Interaction (Second Edition)*, pages 263 – 298. Morgan Kaufmann, Boston, second edition edition, 2017.
- [52] Martin Henschke, Tom Gedeon, and Richard Jones. Touchless gestural interaction with wizard-of-oz: Analysing user behaviour. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction, OzCHI ’15*, pages 207–211, New York, NY, USA, 2015. ACM.
-

- [53] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of oz studies: Why and how. In *Proceedings of the 1st International Conference on Intelligent User Interfaces*, IUI '93, pages 193–200, New York, NY, USA, 1993. ACM.
- [54] Armin Fiedler and Malte Gabsdil. Supporting progressive refinement of wizard-of-oz experiments. pages 62–69, 01 2002.
- [55] J. Isaac. "what is virtual reality?". completegate.com. doi:<https://www.completegate.com/2016070154/blog/virtual-reality-explainedvrdef>. 01 2016.
- [56] D. Coz and D. Henry. Google cardboard. (2014). doi:<https://www.google.com/get/cardboard/>. 01 2014.
- [57] G. Bachelard. *La formación del espíritu científico*. Teoría (Siglo Veintiuno). Siglo Veintiuno, 1993.
- [58]
- [59] M. Gómez-Martín. Aprendizaje basado en juegos. departamento de sistemas informáticos y programación. 01 2003.