# Estrategias para el desarrollo de habilidades de pensamiento computacional: Experiencia de aprendizaje en el Seminario Pensamiento Computacional

Paula Alejandra Rodríguez Chitiva

Licenciatura en Diseño Tecnológico
Departamento de Tecnología
Facultad de Ciencia y Tecnología
Universidad Pedagógica Nacional
2021

# Estrategias para el desarrollo de habilidades de pensamiento computacional: Experiencia de aprendizaje en el Seminario Pensamiento Computacional

Paula Alejandra Rodríguez Chitiva

Linda Alejandra Leal-Urueña Urueña
Directora

Departamento de Tecnología
Facultad de Ciencia y Tecnología
Universidad Pedagógica Nacional
2021

## Resumen

Este informe evidencia la experiencia de aprendizaje obtenida durante el desarrollo del Seminario Electivo Pensamiento Computacional. A través de este curso se hizo un recorrido por los conceptos fundamentales del pensamiento computacional, las técnicas algorítmicas para la solución de problemas, los marcos conceptuales y escenarios pedagógicos para el desarrollo de pensamiento computacional y los enfoques de evaluación del pensamiento computacional. Este informe sintetiza, entre otros aspectos, algunos de los procesos de análisis de información y de estrategias de solución a los problemas y retos propuestos mediante diversas técnicas, conceptos y procesos evaluativos del pensamiento computacional.

La fundamentación y conceptualización fue una fase esencial en el aprendizaje, puesto que ayudan a comprender e interpretar lógicamente cada uno de estos ejercicios, respecto a las estrategias más adecuadas para descomponer un problema, idear diversas estructuras para llevar a cabo una tarea (algoritmos), analizar de qué manera abstraer de una problemática datos esenciales y descartar datos complementarios, idear patrones para resolver una o más situaciones encaminados a diversas disciplinas, entre otros factores que intervienen para lograr soluciones eficaces en distintos campos de la vida del ser humano.

Dicho lo anterior, el propósito es comunicar a la comunidad universitaria ya estudiantes de diversos ciclos de formación, una sistematización de la experiencia en este seminario de maestría, a partir de los resultados alcanzados en cada una de las actividades y que fueron determinantes para aproximarme a los conceptos fundamentales del pensamiento computacional, las técnicas algorítmicas, los escenarios computacionales y los enfoques de evaluación del pensamiento computacional.

# Tabla de Contenidos

## Lista de figuras

## Lista de tablas

# Introducción

De las modalidades de grado para estudiantes de pregrado en la Universidad Pedagógica Nacional está la de cursar materias en la Maestría en Tecnologías de la Información y Comunicación, adscrita al Departamento de Tecnología, de la Facultad de Ciencia y Tecnología. Uno de los seminarios ofertados es el de Pensamiento Computacional en el cual se abordan, entre otros asuntos, sus autores y exponentes, sus técnicas, conceptos y formas de evaluación en el ámbito educativo. Este informe recopila la experiencia de aprendizaje en este seminario a través del abordaje de cada uno de sus módulos.

Los aprendizajes adquiridos en el primer módulo "introducción al pensamiento computacional", propuesto permiten comprender que el pensamiento computacional es un problema complejo que el ser humano debe analizar y comprender, con el objetivo de desarrollar soluciones adecuadas mediante secuencias de instrucciones y algoritmos. Para profundizar este concepto en cada uno de los ejercicios, se analizaron diversas estrategias para organizar los datos y automatizar las soluciones. Las estrategias que se tuvieron en cuenta para lograr la resolución de esta primera fase, estuvieron basadas en los pilares del pensamiento computacional: descomposición, reconocimiento de patrones, abstracción y estructura de algoritmos (Leal-Urueña, 2021), los cuales resultan esenciales para la solución efectiva de los problemas. Finalmente, las soluciones se estructuraron en algoritmos que permiten la solución de los problemas y una mayor comprensión.

El capítulo dos sintetiza la experiencia de aprendizaje en el módulo "técnicas algorítmicas para la solución de problemas" a través del cual pude comprender algunos algoritmos clásicos utilizados en la solución de problemas de búsqueda y ordenación. No obstante su complejidad, es

posible asociarlos con procesos en la vida cotidiana y su representación a través de diversas danzas facilitó su comprensión.

El capítulo tres recoge el análisis de algunos escenarios tecnológicos, que propician el desarrollo de habilidades computacionales, y que han sido diseñados a partir de diversos marcos conceptuales Leal-Urueña (2021). Los aprendizajes alcanzados en este módulo permiten entender el reto en la educación del pensamiento computacional, como un factor preponderante en el desarrollo de habilidades digitales, que no solo están limitadas al estudio de la ciencia de la computación sino que hacen parte de las habilidades de diversas áreas del conocimiento, representadas por el modelo STEAM, pues permiten aplicar diversos procesos de razonamiento a la construcción de conocimiento.

En el capítulo cuatro, se analizan los enfoques de evaluación basado del pensamiento computacional, a partir de varios artículos de investigación, con la finalidad de evaluar diversas alternativas que conduzcan a su comprensión y afianzamiento. Es importante resaltar la aplicación de sistemas de evaluación integrales para evaluar tipos de habilidades de modo que estos procesos fortalezcan y consoliden los aprendizajes en distintos ciclos de educación.

# Capítulo 1

## 1. Fundamentos del pensamiento computacional

Wing (2006), destaca la importancia de vincular el pensamiento computacional en los primeros ciclos de educación (básica primaria y secundaria), para que los estudiantes incorporen en su formación técnicas y metodologías para la solución de problemas en áreas como la lectura, escritura, aritmética, entre otras disciplinas del saber, adecuando este modelo de pensamiento en todos los seres humanos.

A su vez, Sánchez-Vera (2019) menciona que a principios de este siglo se desarrollaron nuevas conceptualizaciones sobre lo que es y lo que implica el pensamiento computacional en relación al desarrollo de sistemas computacionales más intuitivos y la implementación de nuevos modelos de robots para las aulas, el pensamiento computacional se vuelve un referente en el desarrollo de estas actividades dentro de la comunidad educativa y científica.

Por su parte, Leal-Urueña (2021), considera que el pensamiento computacional debe ser considerado una competencia básica del siglo XXI. No obstante, no existe consenso respecto a su conceptualización y a cómo tendría que integrarse en el currículo y trabajarse en el aula, lo que nos lleva a encontrar una gran diversidad de experiencias y enfoques en torno a la temática. El pensamiento computacional, en relación a la tecnología educativa, contribuye a fomentar ideas creativas y didácticas. Para lograrlo es fundamental emplear como primer momento la formación docente el pensamiento computacional para liderar procesos de aula en la actualidad y futuras generaciones. No obstante permitiría el espíritu crítico facilitando la interactividad y el aprendizaje colaborativo en los estudiantes.

El proceso de planificación y consecución de ideas implica procesos de pensamiento computacional, expresados de diferentes modos de análisis. No obstante si se pretende solucionar algún tipo de problemática, se analiza descomponiéndolo en partes más fáciles de comprender, para luego conformar conjuntos de soluciones que conlleven a la solución general, de modo que se pueda comunicar la respuesta adquirida en un conjunto de pasos que puedan ser comprensibles y procesadas por el ser humano y por una computadora.

## 1.1 Pilares del pensamiento computacional

El pensamiento computacional según Leal-Urueña (2020) cuenta con técnicas y habilidades orientadas a la solución de problemas, estos principios son:

### 1.1.1 Descomposición

La descomposición es un principio del pensamiento computacional que tiene como objetivo dividir la problemática en secciones pequeñas, para así analizar cada sección parte por parte, entender lo que implica cada parte en el todo de la problemática, volverlo más comprensible y así llegar a una solución o establecer una serie de soluciones o recomendaciones que sean factibles, se ajuste al análisis realizado y contribuya al objetivo de la problemática. La descomposición en partes reduce la complejidad del análisis y se puede indagar parte por parte tranquilamente y establecer más relaciones que contribuya la solución de la problemática.

### 1.2.1 Abstracción

Los beneficios educativos de poder pensar de manera computacional son varios, empezando por el uso de abstracciones que mejoran y refuerzan las habilidades intelectuales, y que por tanto pueden ser transferidos a cualquier otro ámbito. Además, la abstracción permite

destacar los elementos importantes del problema o situación para así establecer relaciones entre sí y trabajar en la construcción de solución o del elemento que solicite el contexto en el que se desarrolla la problemática.

### 1.3.1 Reconocimiento de patrones

Emplea una misma estrategia para solucionar diversas actividades tanto en el ámbito educativo como en la resolución de problemas en la cotidianidad. El reconocimiento de patrones tiene como función establecer elementos iguales o similares en situaciones para así visualizar cómo pueden intervenir de igual manera en las situaciones que se presenten. En ese sentido, es analizar y establecer los patrones o elementos que surgen de las actividades y que pueden ser funcionales para dichas situaciones.

### 1.4.1 Escritura del Algoritmo

Es un lenguaje formal que, mediante una serie de instrucciones, le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para, de esa forma, crear programas que controlen el comportamiento Físico y lógico de una máquina. La escritura de algoritmos permite tener un orden en las funciones y órdenes de los pasos a seguir y las decisiones que se toman en el proceso y así determinar la mejor manera de abordar la problemática o la solución necesaria.

### 1.2 Aplicación de los conceptos fundamentales del pensamiento computacional orientado a la solución problemas.

En el desarrollo de las temáticas vistas durante el capítulo 1, se describe a continuación las respuestas obtenidas de la sección 1 y actividad 1, basando los aprendizajes obtenidos en el primer módulo: Introducción al pensamiento computacional, describiendo de manera detallada estrategias de solución a distintas modalidades de juego y actividades de razonamiento.

### 1.3 Síntesis Lección 1

En el seminario electivo de pensamiento computacional, referente al primer capítulo Leal-Urueña (2021),“conceptos fundamentales del pensamiento computacional”, se desarrolló en la lección 1 una serie de ejercicios aplicados a la resolución por medio de algoritmos, de manera qué este pilar de “estructuras de algoritmos” enfatiza una secuencia de instrucciones o pasos para llegar a obtener la solución final y posteriormente generar un “patrón general” para dar respuesta al desarrollo de otro tipo de ejercicios.

Respecto al razonamiento sobre el sistema de algoritmos y reconocimiento de patrones se desarrolla un primer ejercicio llamado “el problema del caballo de ajedrez” en el que se tuvo en cuenta 3 tipos de reglas: El movimiento del caballo siempre es en forma de "L", es decir, dos espacios en una dirección seguidos de un movimiento de un cuadrado en ángulo recto, o viceversa.

El movimiento del caballo consiste en un salto a la nueva casilla sin visitar ninguna en el medio, aterrizando siempre en una casilla en el tablero. Encontrar una secuencia de movimientos para que el caballo, comenzando en el cuadro 1, visite todos los cuadros del tablero exactamente una vez y termine nuevamente en el cuadro 1.

Para la realización del algoritmo se estimó un tiempo de 2 horas, en los que se obtuvo los siguientes resultados:

***Tabla 1.1***

Algoritmo de desarrollo de tablero de ajedrez.

| ALGORITMO | |
|---|---|
| Movimiento 1 | Casillas: 1, 4, 5,6 |
| Movimiento 2 | Casillas: 6, 5, 4,8 |
| Movimiento 3 | Casillas: 8, 4, 1, 2 |
| Movimiento 4 | Casillas: 2, 5, 9,10 |
| Movimiento 5 | Casillas: 10, 9,8,4 |
| Movimiento 6 | Casillas: 4, 8, 11,12 |
| Movimiento 7 | Casillas: 12, 9, 8,7 |
| Movimiento 8 | Casillas: 7, 3, 4,5 |
| Movimiento 9 | Casillas: 5, 9, 12,11 |
| Movimiento 10 | Casillas: 11, 8, 4,3 |
| Movimiento 11 | Casillas: 3, 4, 5,9 |
| Movimiento 12 | Casillas: 9, 5, 2,1 |

Fuente: Elaboración propia.

## 1.4 Metodología

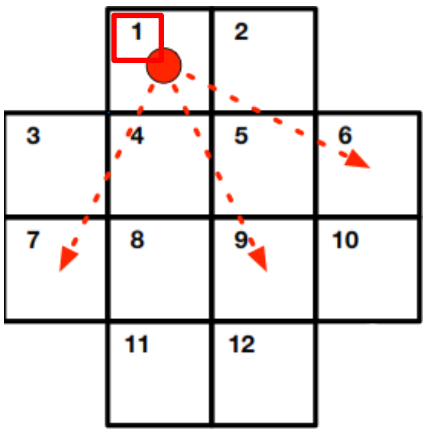
El jugador deberá posicionarse en una de las casillas de requisito, es decir en el recuadro número uno (1) y a partir de este seguir un ciclo, teniendo en cuenta la regla de movimiento del caballo en forma de “L”, desplazar la ficha de modo que encuentre una ruta en la cual no se repita una misma casilla y visite todos los cuadros en el tablero. En caso de evidenciar un movimiento equivocado, de deberá cambiar el ciclo en el transcurso de todo el recorrido.

Para comprender la tabla del algoritmo encontramos un tablero de ajedrez en forma de cruz en el cual consta de 12 cuadros, estos están enumerados previamente del 1 al 12 en la parte superior izquierda, tal y como lo señala la siguiente figura:

### 1.4.1 Casillas

**Figura 1.1**

Casillas tablero de ajedrez.



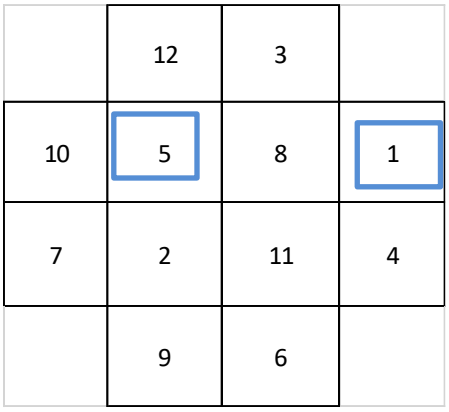

***Fuente:*** Lección 1. Aproximación a los conceptos fundamentales del pensamiento computacional, asignatura pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Para ubicar los movimientos en el tablero de ajedrez, debemos especificar que casillas se tuvo que utilizar aplicando recorrido en forma de “L” y al finalizar este primer movimiento, marcamos en la última casilla del movimiento el número correspondiente a ese recorrido como lo indica la siguiente figura

### 1.4.2 Movimientos

**Figura 1.2**

Movimientos tablero de ajedrez.

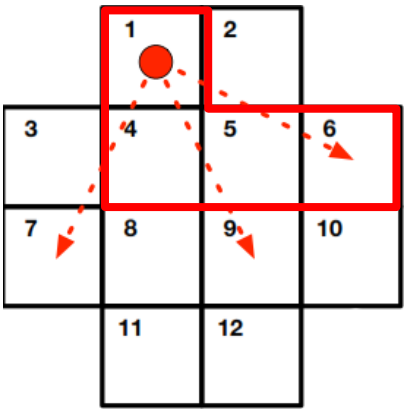



***Fuente:*** Lección 1. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Dicho lo anterior, procedemos a analizar qué proceso se realizó en el movimiento 1, para ello inicia en la casilla 1 para luego dirigirse en dirección hacia la casilla número 4 seguido de las casillas 5 y 6 respectivamente, tal como lo ilustra la siguiente figura:

**Figura 1.3**

Ejemplo Movimiento 1 tablero de ajedrez.

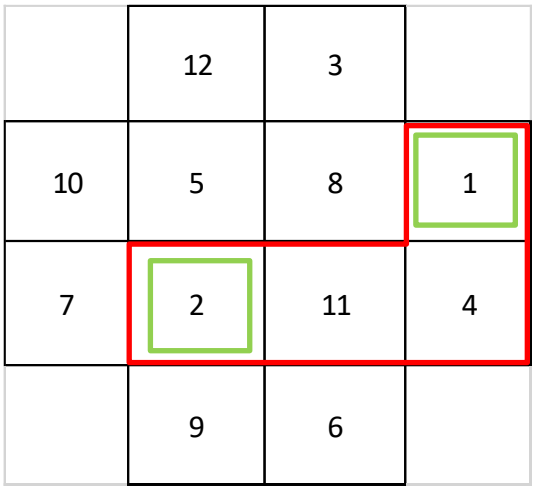



***Fuente:*** Lección 1. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña

No obstante, para realizar el movimiento número 2, como estrategia se debe tomar la última casilla marcada con el movimiento 1 y tomar tal casilla para iniciar una nueva ruta de movimiento y así continuamente con la última casilla, hasta finalizar los 12 movimientos.

**Figura 1.4**

*Ejemplo Movimiento 2 tablero de ajedrez.*




***Fuente:*** Lección 1. Elaboración propia de primer movimiento. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña

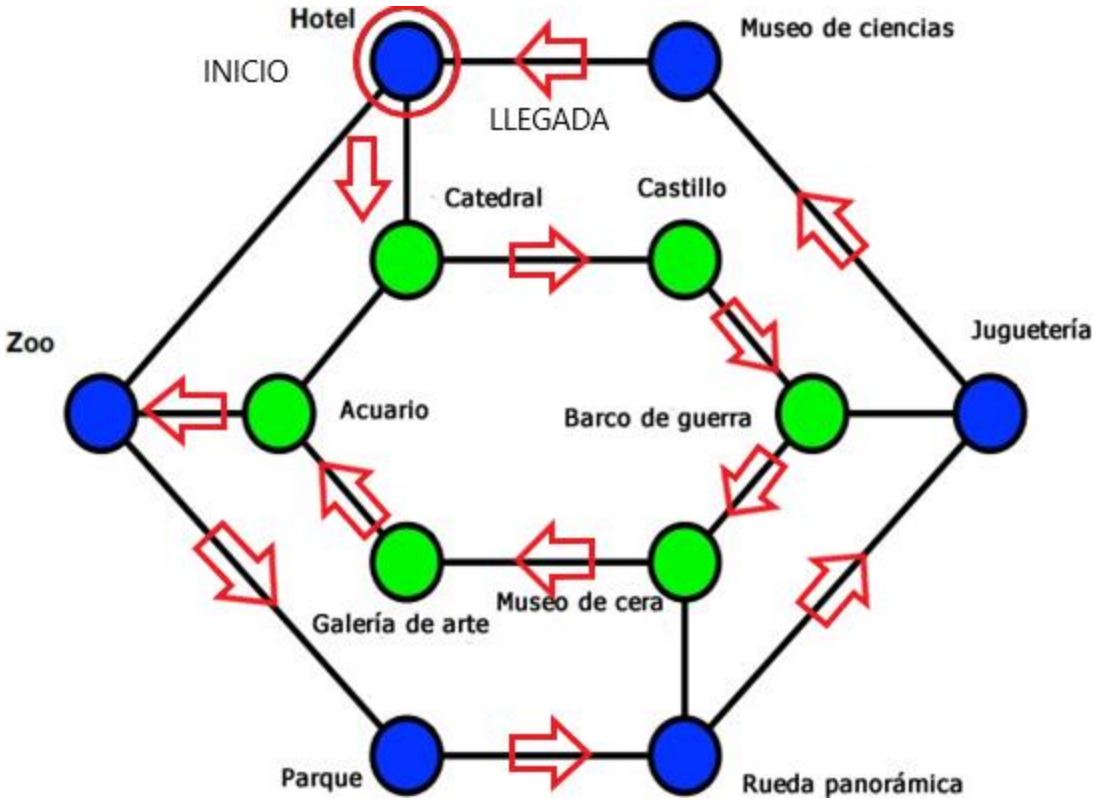
## 1.5 Síntesis 2 actividad problema del guía turístico

En el planteamiento del problema del guía turístico se requiere elegir la ruta más opcional en un mapa de metro, con el objetivo que los turistas en un solo día tengan la oportunidad de visitar todas las atracciones de la ciudad. Para llevar a cabo este reto, iniciamos como punto de partida el "hotel" para luego analizar las conexiones respecto a las líneas de ruta más cercanas, en este caso los lugares marcados con los puntos verdes. Con ello el guía turístico se direccionara hacia la catedral como primer destino y luego dará continuidad de izquierda a derecha respecto al hexágono interno del mapa visitando cada lugar exactamente

una vez, teniendo así un mayor aprovechamiento respecto del tiempo. Al finalizar los trayectos más cortos, el guía turístico se direccionará desde “Acuario” hacia la izquierda y encontrará “Zoo”, uno de los puntos externos del mapa hexagonal señalizado con el punto azul, recorriendo así los demás lugares de destino con mayor trayectoria y retornando nuevamente al destino de partida del hotel en horas de la noche. Para lograr este reto se destinó 1 hora para la construcción de la ruta de direccionamiento, análisis algorítmico del ejercicio y comprobación de la solución en los que se hallaron los siguientes resultados:

**Figura 1.5**

*Mapa de ruta guía turístico.*




***Fuente:*** Lección 1.Elaboracion propia mapa de ruta. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Hotel – Catedral

Catedral – Castillo

Castillo – Barco de guerra

Barco de guerra – Museo de cera

Museo de cera – Galería de arte

Galería de arte – Acuario

Acuario – Zoo

Zoo –Parque

Parque –Rueda Panorámica

Rueda Panorámica –Juguetería

Juguetería – Museo de ciencias

Museo de ciencias- Hotel

Para comprobar si la solución de este problema es funcional se tuvieron en cuenta los siguientes elementos de señalización:

Puntos verdes: Indica menor tiempo de llegada de un lugar a otro.

Puntos azules: Lugares con mayor trayectoria.

Línea de ruta: Especifica el trayecto para llegar a cada lugar, sin la necesidad de pasar al mismo lugar y así poder visitarlos todos en el mismo día.

### 1.5.1 Prueba del algoritmo

Para comprobar la viabilidad un algoritmo, una de las técnicas aplicables en el pensamiento computacional es la descomposición, en el que se divide principalmente el problema en partes más factibles de analizar, para luego recopilar conjuntos de soluciones que conlleven a determinar una solución general.

En el pensamiento propio del ser humano por medio de este tipo de estrategias han estimulado la comprensión de problemas en diversas disciplinas, dando respuesta a las tareas más complejas sin en cuando esta secuencias de instrucciones contribuyan a organizar de manera lógica y coherente cualquier tipo de situación, acción o información.

Por consiguiente para que una problemática se resuelva de manera lógica, es necesario verificar las condiciones que deberán tenerse en cuenta para la solución de una tarea específica, en este caso para el ejercicio del caballo de ajedrez se tuvieron en cuenta los siguientes requerimientos:

1. El punto de partida comienza en la casilla 1
2. Visita todos los cuadros
3. No repetir las casillas ya visitadas por el caballo
4. Termina nuevamente en el cuadro 1.

Se implementan los mismos requisitos en los algoritmos para ambos problemas “tablero de ajedrez” y problema de “guía turístico”, independientemente del contexto donde se desarrolle la dinámica, empleando un mismo patrón, es decir una misma estrategia de solución.

### 1.5.2 Aplicando el pensamiento computacional

El problema del guía turístico fue la actividad más sencilla de realizar, a comparación del tablero de ajedrez, puesto que el muestreo del mapa permitió visualizar un esquema lógico de información, en relación a la ubicación de distintos lugares y las conexiones o líneas de ruta establecidas, sirviendo como una herramienta de apoyo en la resolución del problema inicial donde de evidencian los nodos y las aristas como elementos clave para abstraer la información de manera más sencilla y manejable.

Sin embargo, para aplicar el pensamiento computacional a un tablero de ajedrez posee un grado de complejidad mayor, teniendo en cuenta que su único instrumento o herramienta de apoyo son estrategias y reglas debido a que la valoración posicional de la ficha es fundamental para validar si la dirección en la que vamos es correcta, y si no lo es el jugador deberá planificar una nueva estrategia integrando movimientos cíclicos y al final regresar a la misma casilla de donde se partió, requiriendo durante el proceso de desarrollo tácticas de alto nivel que exigen una alta capacidad de visualización y concentración para lograr este tipo de procesos de razonamiento.

### 1.5.3 Ciclo Hamiltoniano

Leal-Urueña (2021) en la Lección 1 “Aproximación a los conceptos fundamentales del pensamiento computacional” atribuye el principio Hamiltoniano al físico irlandés William Rowan Hamilton, especificando la trayectoria donde se incluye todas las conexiones o aristas de un grafo y a su vez debe pasar exactamente una vez, como también la teoría de grafos de Leonhard Euler matemático suizo afirma que si un grafo posee más de dos vértices impares, no existe una trayectoria.

Dicho lo anterior, las dinámicas del caballo de ajedrez y el guía turístico emplean un mismo patrón de solución, por tanto el siguiente reto consiste en realizar un grafo similar al del guía turístico, pero en esta ocasión para el caballo de ajedrez de modo que facilite su modo de entendimiento (abstracción) y solución, para ello se encontraron los siguientes resultados:

Respuesta

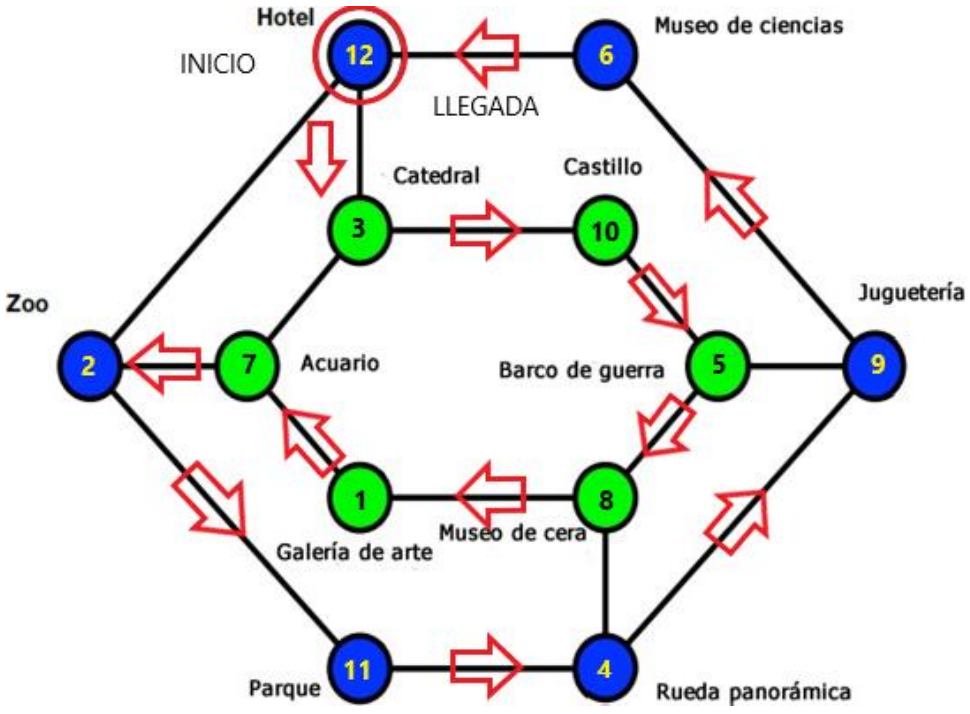
¿Cuáles soluciones a este problema puedes encontrar utilizando tu grafo?

Encontrar el orden de los números correspondientes a cada casilla en el tablero de ajedrez.

2. Visualizar una secuencia de pasos guía para resolver el ajedrez.

**Figura 1.6**

*Grafo de solución tablero de ajedrez.*




***Fuente:*** Elaboración propia grafo de ajedrez, Lección 1. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Luego de construir un grafo se conforma una tabla logarítmica, identificando el número de casillas que posee el tablero de ajedrez, es decir 12 casillas y en cada una de ellas ubicar las respuestas obtenidas por los movimientos del caballo. Por medio de esta técnica de solución por medio de nodos y aristas en un mapa brindaría una respuesta inmediata a este ejercicio, como se muestra en la siguiente tabla:

***Tabla 1.2.***

Tabla de algoritmo tablero de ajedrez.

| Algoritmo de solución tablero de ajedrez | |
|---|---|
| Casilla 1 | #12 |
| Casilla 2 | #3 |
| Casilla 3 | #10 |
| Casilla 4 | #5 |
| Casilla 5 | # 8 |
| Casilla 6 | #1 |
| Casilla 7 | # 7 |
| Casilla 8 | # 2 |
| Casilla 9 | #11 |
| Casilla 10 | # 4 |
| Casilla 11 | # 9 |
| Casilla 12 | #6 |

***Fuente:*** Elaboración propia, descripción general de algoritmo de tablero de ajedrez, Lección 1. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 1.5.4 Mismo problema, misma solución

Con lo aprendido en el desarrollo de grafos, aplicamos un pilar esencial del pensamiento computacional conocido como generalización, identificando problemas con estructuras similares e idear una estrategia de reglas o patrones más generales dando respuesta a 2 o más contextos.

Como próximo ejercicio se requiere encontrar correspondencia entre los lugares turísticos y los números del tablero de ajedrez y se obtuvieron las siguientes similitudes:

**Tabla 1.3**

*Similitudes lugares turísticos y Tablero de ajedrez.*

| LUGARES TURISTICOS | TABLERO DE AJEDREZ |
|---|---|
| Existen 12 lugares | Existen 12 números |
| Indicar la ruta para visitar secuencialmente todos los lugares | Indicar la ruta para ubicar de manera ordenada o secuencial en las casillas de la tabla |
| No se repiten los sitios turísticos | No se repite los números en las casillas |
| Siguiendo la ruta, siempre se llega a un punto de partida | Siguiendo las reglas, termina en el punto de partida. |

***Fuente:*** Elaboración propia, descripción sobre las similitudes de actividad 1 y actividad 2, lección 1. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Para el desarrollo de esta actividad se identificó elementos comunes teniendo en cuenta los siguientes factores:

1. La cantidad de casillas en el tablero de ajedrez y la cantidad de sitios turísticos que se observaron en el mapa de ruta del metro.
2. A través de la técnica del grafo se diseñó un sistema para organizar secuencialmente una determinada cantidad de datos, facilitando su lectura y solución.
3. Aplicamos el Ciclo Hamiltoniano respecto a la no repetición a través de grafos y datos referentes al número de casillas y sitios turísticos.
4. Ambos tipos de ejercicios disponen de un requerimiento de volver al punto de partida, para el guía turístico finalizar una ruta en el hotel y para el tablero de ajedrez realizar movimientos cíclicos para finalizar en la casilla 1.

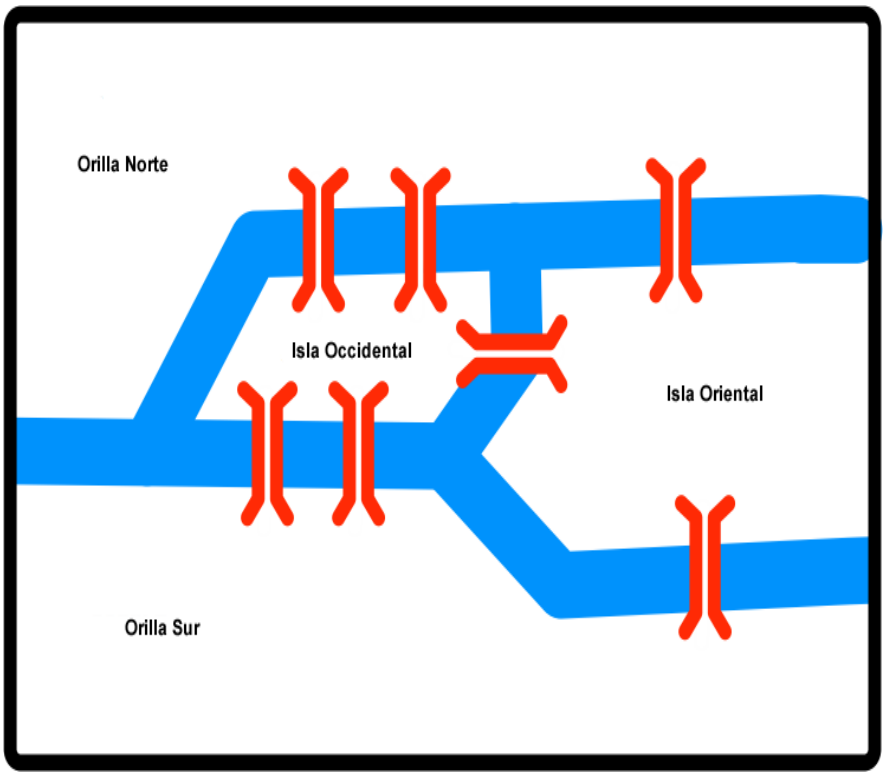
### 1.5.5 El problema de los puentes

Se muestra un mapa de la ciudad de Königsberg, identificando que un río que atraviesa el centro de la ciudad, sus dos islas y los siete puentes que cruzan el río, se plantea idear una ruta para que visite cada parte de la ciudad (las orillas norte y sur y ambas islas) y que cruce cada puente una vez, la ruta debería comenzar y terminar en el mismo lugar.

.

**Figura 1.7**

*Mapa de la ciudad de Königsberg.*




***Fuente:*** Mapa ciudad de Königsberg, imagen *de* lección 1. Aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña

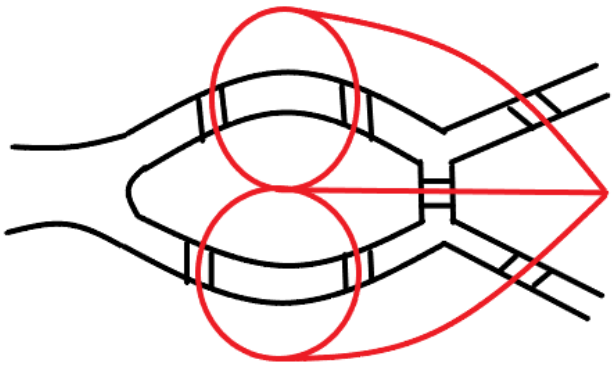
En el documento (Patiño Avendaño & Charry, 2013) en teoría de grafos explica que es un enigma histórico resuelto hace 300 años, donde pregunta cuánto tiempo llevaría recorrer los puentes de Königsberg, respondiendo Leonhard Euler (1707-1738), matemático suizo, no ser posible realizar un solo recorrido pasando por los 7 puentes una sola vez por las siguiente razón:

Si el grafo está caracterizado por 7 puentes es decir números de grado impar no se puede realizar una sola trayectoria, puesto que tendrá que volver al punto inicial cruzando uno de los puentes dos veces. Solo se cumpliría si el mapa contara con 4 puentes de ida y 4 puentes de regreso para dar cumplimiento a la condicional de realizar un solo recorrido por la ciudad.

A continuación veremos un borrador sobre la ruta en la ciudad, para intentar resolver este enigma:

**Figura 1.8**

Ruta puentes de Königsberg.



***Fuente:*** Elaboración propia de ruta de la ciudad de Königsberg, lección 1, aproximación a los conceptos fundamentales del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 1.5.6 Poniendo en juego tu comprensión del pensamiento computacional

En la actividad 1 propuesta por Leal-Urueña, (2021) “Poniendo en juego tu comprensión del pensamiento computacional” consiste en poner en práctica los diversos pilares del pensamiento computacional, para ello encontramos una sopa de letras donde se destacan algunos científicos que hicieron parte de la historia de la ciencia de la computación.

Luego de encontrar solución a esta actividad se describe de forma secuencial los pasos que se tuvieron en cuenta para encontrar los nombres en el listado, indicar que tipo de pilares o elementos fueron esenciales para el desarrollo de habilidades específicas (abstracción, descomposición, generalización, evaluación y diseño algorítmico) y por ultimo generalizarlos por medio de un algoritmo legible para que pueda ser comprensible para una computadora o modular, para la resolución de otro tipo de situaciones en la lógica del ser humano. Los datos adquiridos en el desarrollo de esta actividad son las siguientes:

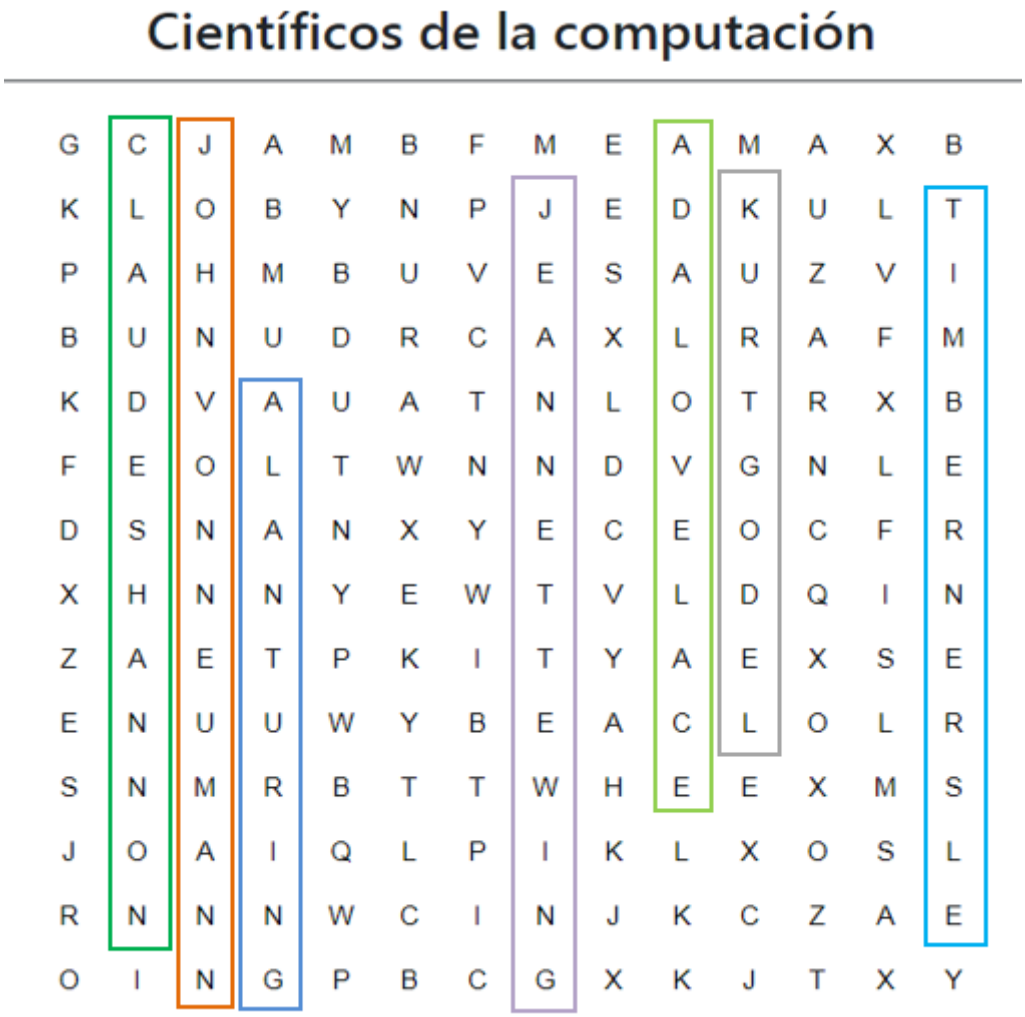
### 1.5.7 Científicos de la computación

Listado de palabras

Ada Lovelace - Jeannette Wing -Alan Turing - John Von Neumann -Tim Berners Lee - Claude Shannon –KurtGodel.

**Figura 1.9**

Solución sopa de letras.




***Fuente:*** Elaboración propia, sopa de letras sobre destacados científicos de la computación. Actividad 1. Poniendo en juego tu comprensión del pensamiento computacional.Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Descripción del proceso

**¿Que buscar?**

-Listado de palabras ocultas en la sopa de letras.

**¿Cómo saber si encuentro lo que busco?**

-Observar la primera letra de la palabra que desea encontrar y busque solamente esa letra en la sopa de letras.

-Al encontrar la primera letra de la palabra, buscar si la dirección de la palabra completa se encuentra en dirección horizontal, vertical o diagonal.

**¿Cuándo detenerme?**

- Cuando encuentre la palabra deberá encerrarla o marcarla con el fin de comprobar que la palabra fue encontrada.

2. Sobre el texto de tu descripción identifica si aplicaste procesos descomposición generalización, abstracción y construcción de algoritmos.

*Generalización (Reconocimiento de patrones)*

Al reconocer patrones, podemos hacer predicciones de lo que sucederá después, crear reglas y solucionar otros problemas. En la sopa de letras se reconoce de manera general las siguientes ideas:

1. Las palabras pueden estar ubicadas en cualquier dirección (Vertical, horizontal o diagonal) o en sentido derecho o al revés.

2. Identificar las letras que componen la palabra y su orden respecto a la forma de escritura.

*Algoritmos*

Se maneja una secuencia exacta de instrucciones o secuencias para realizar una sopa de letras, estas pueden ser:

1. Está compuesto por un tema central
2. A partir de la temática central se deriva el listado de palabras a encontrar.
3. Visualizar un conjunto de letras que están distribuidas en un tablero de manera horizontal y vertical.
4. Busca la primera letra de la palabra en sentido horizontal, vertical, diagonal, al revés al derecho hasta encontrar la palabra completa.
5. Encerrar en un círculo ovalado la palabra encontrada para distinguir del resto de letras en el tablero.
6. Tachar la palabra encontrada en el listado de palabras, con ello no se tendrá que buscar nuevamente.

*Abstracción*

Cuando existe una concentración en un conjunto pequeño de letras adyacentes, la atención se convierte en un factor de percepción importante, evidenciando con mayor claridad información relevante como los datos que queremos encontrar y excluyendo aquellos elementos que se encuentran desorganizados dentro de una lógica de juego.

3. Escribe tu algoritmode solución:

Al derecho (normal)

Al revés

Hacia arriba (vertical)

Hacia abajo (vertical)

Diagonal Superior Izquierda

Diagonal Inferior Izquierda

Diagonal Superior Derecha

Diagonal Inferior Derecha

### 1.5.8 Bloques cortados

Vilchez (2017) , resalta la importancia de resolver diversos tipos de problemas a través de una metodología, mediado por una serie de acciones o situaciones para conseguir un objetivo. Por tanto los algoritmos son una herramienta de apoyo para comprender más fácilmente los resultados.

Estos algoritmos están caracterizados por ser concisos, ordenados y a su vez conformen tipos de reglas que puedan ser establecidas para facilitar su comprensión. En esta última actividad se pretende solucionar un rompecabezas de bloques cortados haciendo uso de 2 tipos de reglas para su solución las cuales fueron las siguientes:

1. Cada área demarcada por las líneas más oscuras debe contener los números del 1 al número de cuadrados en el área. Por ejemplo, el área superior en el primer rompecabezas consta de 5 cuadrados, por lo que esos cuadrados deben llenarse con los

números: 1, 2, 3, 4 y 5, sin números repetidos. Si el área tiene dos cuadrados, como el de abajo a la izquierda, del mismo rompecabezas, debe llenarse con los números 1 y 2.

2. Ningún número puede estar al lado del mismo número en cualquier dirección, ya sea horizontal, vertical o diagonal. Entonces, en la cuadrícula de abajo, el hecho de que haya un 4 en el costado significa que no puede haber un 4 en ninguno de los 5 cuadrados que lo rodean.

Descripción del proceso

1 paso: Se toma el área con menor número de casillas, si son dos casillas en blanco se completa con los números 1 y 2.

2 paso: Enumerar el área según la cantidad de casillas, por ejemplo si hay 5 casillas se enumera del 1 al 5.

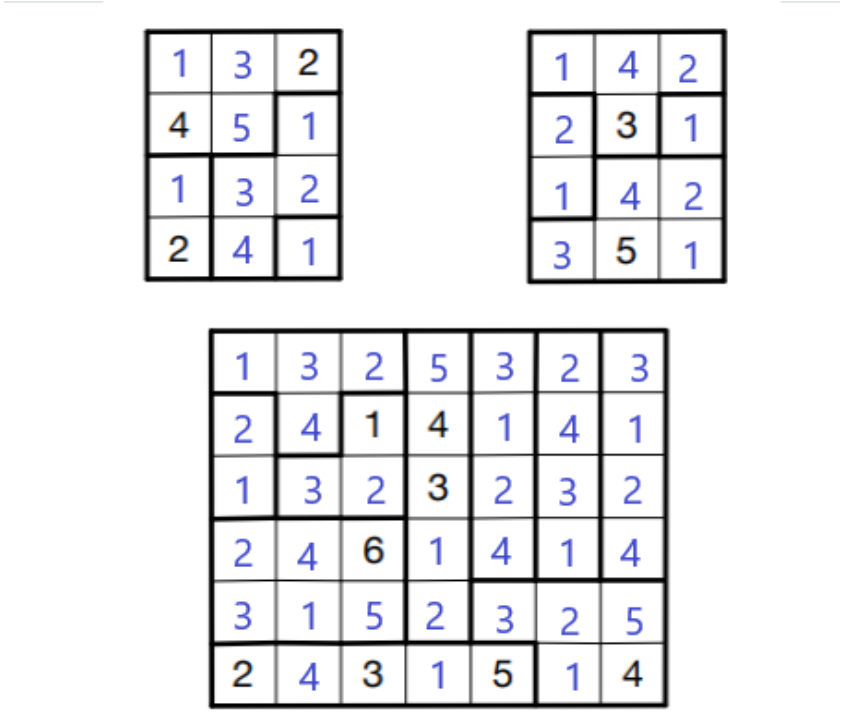
3 paso: Si el área contiene números fijos, no se deberá repetirse el mismo número.

4 paso: Tanto en las columnas como en las filas no debe repetirse el mismo número.

5 paso: Se recomienda ubicar el número 1 para cada área, luego el número 2 y así sucesivamente hasta rellenar en su totalidad el tablero.

**Figura 1.10**

*Solución de rompecabezas bloques cortados.*




**Fuente:** Elaboración propia de rompecabezas de bloques cortados.Actividad 1. Poniendo en juego tu comprensión del pensamiento computacional.Asignatura pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

2. Sobre el texto de tu descripción identifica si aplicaste procesos de: descomposición, generalización, abstracción y construcción de algoritmos:

Descomposición: Tomar área por área por separado para resolverlo de manera más rápida y fácil.

Algoritmos: Llevar a cabo las reglas que imparte el juego del sudoku rompecabezas.

Generalización: Detectar las similitudes entre áreas para así poder ordenar los números de manera correcta sin que se repitan en ninguna de las direcciones (horizontal y vertical).

3. Escribe el algoritmo de tu solución. No olvides probarlo.

Regla 1: Completar las casillas vacías con un solo número del 1 a la cantidad de espacios por área.

Regla 2: en una misma fila no puede haber números repetidos

Regla 3: en una misma columna no puede haber números repetidos

Regla 4: en una misma región no puede haber números repetidos

# Capítulo 2

## 2. Técnicas algorítmicas para la solución de problemas

En el presente capítulo, se presenta la implementación de dos tipos técnicas clásicas para la solución de cada una de las actividades de búsqueda y ordenación, visto desde un lenguaje de programación (diagramas de flujo) hasta la cotidianidad de las personas organizando alfabéticamente una secuencia de números por medio de una danza.

En el desarrollo de la lección 2 y lección 3 se tuvo en cuenta varios criterios de organización como la ubicación de valores numéricos de menor a mayor o decrecientemente, como también identificar la posición de un valor respecto a un conjunto de datos que buscan una secuencialidad o linealidad. En los diversos bailes, los algoritmos de búsqueda fueron más evidentes, puesto que la idea del bailarín siempre fue encontrar un elemento de la lista de idear una secuencia de pasos para: encontrar la pareja con el mismo dato numérico, realizar comparaciones para generar una relación de orden menor (<) y mayor (>) o de acuerdo a ciertos parámetros organizar un conjunto datos.

Dicho lo anterior, la conceptualización del algoritmo como lo plantea la lección 2, crea procesos por medio de requerimientos para procesar una tarea de inicio a fin, descomponiendo el elemento de estudio en múltiples tareas para su ejecución, de manera que este tipo de ejercicios sean fácilmente descriptivos dentro de una secuencia. También fue esencial identificar en qué parámetros se justifica el ejercicio, si es necesario realizar una acción en un número mínimo de pasos o si depende del grado de dificultad, se extenderá el conjunto de pasos hasta encontrar una solución en una búsqueda binaria a través de diagramas de flujo, comúnmente utilizados en la programación.

Teniendo en cuenta el análisis de un problema a través de búsqueda binaria se realizaron 2 tipos de actividades basados los procesos que realiza un diagrama de flujo, donde el algoritmo cuenta con un inicio, una acción, una condicional y finalmente un acierto o desacierto de acuerdo a la elección del sujeto. No obstante, la actividad consistió en adivinar un número entre un rango entre dos números y a partir de este analizar que estrategias serían las más apropiadas para lograr acertar en un número finito de pasos.

## 2.1 Algoritmos de búsqueda

Los algoritmos de búsqueda consisten encontrar o recuperar datos o valores numéricos por medio de estrategias más eficientes dentro de una sucesión ordenada o desordenada, estas estrategias están mediadas por dos tipos de búsqueda:

### 2.1.1 Búsqueda aleatoria

Consiste en adivinar un dato numérico entre dos rangos (valor mínimo y valor máximo) tomando como referencia un diagrama de flujo, en el que representa un estado de inicio, una pregunta sobre el número que desea elegir y por ultimo un condicional que determina si adivina el número al azar, en caso de no acertar el ciclo vuelve a repetirse generando la misma pregunta hasta lograr la respuesta correcta.

### 2.1.2 Búsqueda binaria

La búsqueda binaria es una estrategia para encontrar un elemento en una lista ordenada de datos, tomando un elemento para luego dividirlo repetidamente a la mitad de la lista que podría contener al elemento. Uno de los ejemplos realizados en esta sección, consistió en adivinar un número al azar dentro de un conjunto finito de pasos para su solución o emplear una estrategia a

través de la danza con el fin de evidenciar cuantas comparaciones fueron posibles para encontrar la pareja con el mismo dato numérico.

Por otro lado, existen varios métodos de ordenación, unas más dispendiosas que otras pero conllevan a generar listas de manera intuitiva para organizar la información y solucionar planteamientos de una manera más eficiente, estos están divididos en:

### 2.2 Algoritmos de ordenación

Leal-Urueña (2021) plantea en el módulo 2, introducción a las técnicas algorítmicas, varios tipos de algoritmos clásicos de ordenación: la ordenación por selección y la ordenación por burbujas son buenos puntos de partida, aunque también analizaremos la ordenación rápida (quicksort) y la ordenación combinada (mergesort) para observar que tipos de algoritmos son más eficientes que otros en la solución del mismo problema.

Conceptualiza a los algoritmos de ordenación como una conformación de un conjunto de elementos y criterios que deben ubicarse según un carácter numérico o alfabético en los que se describen varios tipos de algoritmos de ordenación, descritos por Leal-Urueña (2021)

#### 2.2.1 Ordenamiento por selección

En la programación a medida que se traslada un elemento se lleva a la posición correcta, como se formuló en la estrategia de la danza donde se localiza el menor valor numérico en la primera posición, luego se ubica el valor con un grado mayor y así hasta llegar al último elemento conformando una secuencia alfanumérica.

#### 2.2.2 Ordenamiento por inserción

Consiste en tomar dos elementos numéricos y realizar una comparación de dos valores para determinar quien ocupa ese espacio, por ejemplo 2 es mayor que 1, por tanto quien ocupará la

posición correcta será dato con menor valor numérico y así sucesivamente ir determinando por medio de la estrategia (< ; > ) una escala ascendente o descendente según los criterios iniciales.

### 2.2.3 Ordenamiento burbuja

Consiste en comparar un dato con su sucesor y si determina que está en una casilla que no corresponde le cederá el espacio para conformar un ordenamiento total de los datos, no es un factor determinante si realiza un valor impar intercambio con su sucesor par.

### 2.2.4 Ordenamiento shell

Es una estrategia similar al ordenamiento por inserción y consiste en ordenar subconjuntos de elementos saltando uno o más números hasta que todos los elementos se encuentren organizados.

### 2.2.5 Ordenamiento por heapsort

El método Heapsort es una estrategia que trabaja con árboles estableciendo una secuencia de vectores donde en la primer secuencia estarán ubicados los datos mayores y a medida que se va cambiando de posición en el árbol, se van ubicando los números en el listado y posteriormente se va trasladando el valor numérico mayor hasta la última casilla.

### 2.2.6 Ordenamiento por quicksort

Un listado de números de divide en dos grupos respectivamente, uno de estos elementos constituidos por datos de menor valor y el segundo grupo conformándose por los datos de mayor dato numérico, como segundo paso se organiza cada grupo en una escala ascendente y finalmente se reúnen estos dos grupos para conformar nuevamente un solo listado dentro de la secuencia.

## 2.3 Aplicación de Técnicas algorítmicas

En el desarrollo de las lecciones 2 y 3 se logró un aprendizaje respecto a los fundamentos de la estructura de datos y la funcionalidad que se le ha dado a distintos tipos algoritmos: búsqueda y ordenamiento para resolver problemas sencillos hasta grados más complejos.

En la aplicación de las técnicas algorítmicas se observará a través de los siguientes ejercicios los planteamientos junto al análisis de los procesos para llegar a una solución general, haciendo uso de los diferentes tipos de estrategias de ordenamiento de datos y búsqueda lineal los cuales han sido elementos representativos en la programación estructurada.

### Síntesis lección 2

## 2.4 Introducción a las técnicas algorítmicas

Pregunta 1

El video ilustra, a través de un baile, el algoritmo de búsqueda lineal

Respuesta 1

En la descripción del video se presentan las siguientes comparaciones:

El bailarín #7 danza con la bailarina #3

El bailarín #7 danza con la bailarina #9

El bailarín #7 danza con la bailarina #0

El bailarín #7 danza con la bailarina #5

El bailarín #7 danza con la bailarina #8

El bailarín #7 danza con la bailarina #7

CONCLUSIÓN: Se presentaron 6 comparaciones para que el bailarín encontrara la pareja con el mismo número

Pregunta 2

Observa en qué consiste la búsqueda binaria, explica con tus palabras las comparaciones que se realizan en el baile.

Respuesta 2

1. Encontramos inicialmente 7 bailarinas, cada una cuenta con un espacio del X(0) a X(6).
2. El bailarín Se dirige al espacio X (3) encontrando a la bailarina 5, descartando los valores menores a este.
3. El bailarín Se dirige al espacio X (5) y danzando encuentra la bailarina número 8, girando a alejan estos dos números en un grupo al ser valores mayores a 7.
4. El bailarín Retrocede un espacio hacia X (4) y se observan los mismos pasos del bailarín con pareja con el mismo número.

Pregunta 3

Después de unos pocos intentos, o tal vez simplemente razonando lógicamente sobre los diagramas de flujo, debería ser evidente que los dos primeros métodos son mucho más lentos que el tercero, a menos que se tenga mucha suerte. Si estamos tratando de adivinar un número, digamos entre 0 y 127, los dos primeros algoritmos tomarían en promedio 64 pasos para

encontrar el número, a veces más, a veces menos. Si usamos el algoritmo de búsqueda binaria, lo encontraremos en solo siete intentos ¡Pruébalo! Y explica tu razonamiento

Respuesta 3

Se toma el rango de números entre 0 y 127

1) 127 dividido en 2 grupos 63+64 =127

2) El número esta entre el rango menor entre el 0 y 64? SI

Límite inferior 0 < 64

3) El número esta entre el rango mayor entre el 64 y 127? NO

Límite superior 64 > 127

4) Si está en el rango MENOR divido nuevamente = 32

5) Si está en el rango MAYOR divido nuevamente = 32

6) Establezco que el número es < ó > que 32? MENOR

7) El número es la medio del rango menor = 16 Adivinando el número basado en el diagrama de flujo de búsqueda binaria.

Pregunta 4

Las ganancias de eficiencia se vuelven aún más dramáticas con un rango más grande de números iniciales: adivinar un número entre uno y un millón tomaría un promedio de 500.000 con cualquiera de los dos primeros algoritmos, pero reduciríamos el número a solo 20 pasos con búsqueda binaria.

Respuesta 4

Se toma el rango de números entre 0 y 1. 000.000

1) 127 dividido en 2 grupos 500.000 y 500.000

2) El número esta entre el rango menor entre el 0 y 500.000?

Límite inferior 0 y 500.000

3) El número esta entre el rango mayor entre 500.000 y 1.000.000?

Límite superior 500.000 y 1.000.000

4) Si está en el rango MENOR divido nuevamente = 250.000

5) Si está en el rango MAYOR divido nuevamente = 250.000

6) El número esta entre el rango entre 0 y 250.000?

Límite inferior 0 >250.000

7) El número esta entre el rango entre 250.000 y 500.000?

Límite superior 250.000 >500.000

8) Si está en el rango MENOR divido nuevamente = 125.000

9) Si está en el rango MAYOR divido nuevamente = 125.000

10) El número esta entre el rango menor entre 0 y 125.000?

Límite inferior 0 >125.000

11) El número esta entre el rango mayor entre 125.000 y 250.000?

Límite superior 125.000 >250.000

12) Si está en el rango MENOR divido nuevamente = 62.500

13) Si está en el rango MAYOR divido nuevamente = 62.500

14) El número esta entre el rango menor entre 0 y 62.500?

Límite inferior 0 >62.500

15) El número esta entre el rango mayor entre 62.500 y 125.000?

Límite superior 62.500 >125.000

16) Si está en el rango MENOR divido nuevamente = 32.000

17) Si está en el rango SUPERIOR divido nuevamente = 32.000

18) El número está entre el rango menor entre 0 y 32.000?

NO

Limite menor 0>32.000

19) El número esta entre la mitad del rango mayor entre 32.000>62.500?

SI

Límite superior 32.000 >62.500

20) El número es 16.000

Pregunta 5

¿Cuáles serían las técnicas de búsqueda, más eficientes, para encontrar el estudiante más joven en una clase o para determinar la ciudad de origen de la mayoría de los estudiantes en una clase?

Explica tu respuesta y explica cómo aplicarías el (o los) algoritmo(s) que hemos aprendido para solucionar estos problemas.

Respuesta 5

Para encontrar el estudiante más joven de una clase, se tienen en cuenta los Siguientes datos:

1) 30 estudiantes se dividen en 2 grupos de 15 personas

2) De esas 15 personas para ambos grupos hay rangos de edades de 7, 8, 9 y 10

3) Un grupo selecciona la persona con edad menor

4) El segundo grupo selecciona la persona con edad menor

5) Se comparan ambos datos de edad ><

6) Se obtiene el estudiante más joven

Para determinar la ciudad de origen de la mayoría de los estudiantes se determinaron los siguientes datos:

1) 30 estudiantes dividen en 2 grupos de 15 personas

2) De esas 15 personas para ambos grupos hay rangos de diferentes ciudades de origen

3) 1 grupo selecciona las personas con la misma ciudad de origen

4) El 2 grupo selecciona las personas con la misma ciudad de origen

5) Se comparan ambos resultados de los grupos definiendo un dato numérico

6) Se obtiene la ciudad de origen de la mayoría de los estudiantes.

## Síntesis lección 3

### 1.5 Algoritmos de ordenación

Ordenación por selección

Pregunta 1

¿Cuántas comparaciones son necesarias para ordenar una lista de 10 elementos por selección?

Respuesta 1

Basado en el video, 45 comparaciones son necesarias para ordenar una lista de 10 números 9+8+7+6+5+4+3+2+1 = 45 elementos

Ordenación Burbuja

Pregunta 2

Siguiendo el algoritmo de ordenación de burbuja ¿cuántas comparaciones hacen falta para organizar una secuencia?. Ilustra el ordenamiento de burbuja con una danza húngara.

Respuesta 2

Hacen falta 26 comparaciones, para ordenar una lista de 10 elementos teniendo en cuenta:

-9 veces para organizar los números 8 y 9.

- 7 veces se organizó el número 7.

-6 veces se arreglaron los elementos 6 y 5.

- En la 4 comparación se organizaron los números 4, 3, 2,1, 0.

-Dando una suma de 9+7+6+4 =26 comparaciones

Ordenación por Inserción

Pregunta 3

Explica con tus palabras la secuencia de pasos que se sigue en este algoritmo.

Respuesta 3

Secuencia de pasos:

1. Las posiciones van del 0 al 9

2. La secuencia desorganizada de números es 3, 0, 1, 8, 7, 2, 5, 4, 9,6.

3. En la primer posición se comparan dos cantidades el número "3" y el "0".

4. Ocupará la posición A (0) el número menor entre los dos, es decir cero y la posición A (1) encontrar un número menor a este.

5. Así sucesivamente cada posición A (2) hasta la A (9) se compararán de menor a mayor en número menor entre dos cantidades.

6. Se compara cada número con la posición anterior, si es mayor que su antecesor, no se traslada de posición

7. Finalmente los números se verán organizados de 1, 2, 3, 4, 5, 6, 7, 8,9.

Ordenación por mezcla (Mergesort)

Pregunta 4

¿Cuántas comparaciones necesitaron esta vez los bailarines para ordenar los 10 elementos?

Respuesta 4

Se necesitaron 27 comparaciones para ordenar los 10 elementos, entre grupos pares e impares.

Ordenación mediante montículos (Heapsort)

Pregunta 5

*Explica con tus palabras, los pasos que se siguen en este algoritmo de ordenación.*

Respuesta 5

Secuencia de pasos:

1. Se tiene en cuenta las posiciones del A (1) hasta A (7) donde se establecerá un orden de números.

2. Se maneja una jerarquía donde los triángulos hacen referencia a: Los números se organizan del triángulo se posicionan los números menores y en la punta superior del triángulo los números mayores.

3. Se realiza una comparación entre los números base del triángulo en este caso A (4) y A (5) mayor, igualmente se realiza la misma operación con A (6) y A (7).

4. Ya teniendo claridad de cuál es el número mayor se compara con las puntas superior del que cantidad es mayor.

5. Solamente ocupara la posición A (3) y A (2) los números mayores de la bases de los triángulos

6. Los números de la posición A (7) y A (1) se comparan validando el mayor de la secuencia del listado.

7. Los números de La posición A (2) y A (3) puntas superiores del triángulo se comparan identificándose la posición A (1) que es la cima del triángulo ocupándolo el mayor de estas dos posiciones.

8. El número de la posición A (6) pasa a la casilla A (1) ubicando el número 6 al listado de números se ubicaran de forma descendente.

9. El número de La posición A (5) pasa a la posición A (1) desplazando el número 5 al listado de números.

10. Se emplea la misma dinámica de comparación de números mayores y menores hasta conformar una secuencia de mayor a menor 9, 8, 7, 6, 5, 4, 3, 2,1.

Ordenación rápida de Hoare (Quicksort)

Pregunta 6

¿Cuántas comparaciones se necesitan para ordenar los 10 elementos? Explica tu respuesta.

Respuesta 6

Se requiere 21 comparaciones para ordenar 10 elementos, tomando un número por ejemplo el #3 y se realiza una comparación con números menores, organizándose respectivamente en el listado de números. Si el número es mayor del asignado sigue ocupando la misma posición.

Se divide en dos grupos para organizar los números más pequeños y en otra los números de mayor grado, del segundo grupo se compara dos cantidades de acuerdo a las posiciones ubicadas de A4, Se realiza la misma operación, identificando el número mayor entre ambos y ocupará la casilla el número de menor valor, en ese sentido encontraremos ya los números organizados.

Ordenación por incrementos (Shellsort)

Pregunta 7

Como reto final, para poner a prueba el desarrollo de tu pensamiento algorítmico, explica el algoritmo para ordenar los elementos.

Respuesta 7

1. El número 2 de la posición A (5) pasa a posición a (0) ya que es un valor menor que el número 3 y este se traslada a la posición A (5).
2. El número 5 posición A (6) se compara con 0 pero al ser un número mayor se mantiene en su mismo espacio.
3. El número 4 posición A (6) se compara con 1 pero al ser un número mayor se mantienen en la misma posición.

4. El número 8 posición A(3) se compara con número 9 posición A(8) pero al ser número menor se mantiene en su mismo espacio.
5. El número 7 posición A (4) se compara con número 6 posiciones A (9) y al ser el número 6 menor intercambia de posición.

Conclusión

Se verifica siempre el número mayor entre ambos, organizando la secuencia de números de menor a mayor y de mayor a menor.

# Capítulo 3

## 3. Escenarios computacionales para el desarrollo del pensamiento computacional

El pensamiento computacional debería considerarse una de las competencias predominantes en el ámbito educativo y en la sociedad, con el objetivo de reforzar aquellas habilidades duras para la solución de problemas complejos. En el campo de las ciencias de la computación se han desarrollado aportes significativos en el campo de la inteligencia artificial, el modelado y la simulación, favoreciendo las grandes industrias, la medicina, la arquitectura entre otros escenarios, que sustentan sus avances por medio de soluciones innovadoras y de representación.

Sin embargo, en los primeros ciclos de educación es importante que el maestro incorpore en su proceso de enseñanza y aprendizaje estrategias donde vincule secuencias lógicas, abstracción, generalización y la construcción de patrones que el estudiante logre comprender, impulsando una mayor confianza en sus procesos de aprendizaje en distintas áreas, como lo propone el modelo STEM (ciencia, tecnología, ingeniería, matemáticas).

La enseñanza del pensamiento computacional refuerza y estructura de mejor forma los conocimientos que se adquieren en la escuela (Edacom tecnologia educativa , 2020) entendiéndose que es una manera de aprender por medio de la práctica en los estudiantes y posteriormente lo practiquen a lo largo de su proceso formativo.

Autores como Wing (2006), Aho (2012), Grover & Pea (2018), Weintrop et al (2016) y la Royal Society (2014) han construido conceptos y prácticas que propicien la educación en el pensamiento computacional, una de las estrategias es hacer uso de los recursos informáticos para

dominar el lenguaje de la programación, estos escenarios pueden ser actividades conectadas o desconectadas que se pueden utilizar a la hora de crear, solucionar y compartir proyectos.

### 3.1. Descripción del escenario computacional

La dinámica para el capítulo 3 consistió en analizar los distintos escenarios tecnológicos, útiles para el desarrollo del pensamiento computacional, tomando como referencia los marcos conceptuales brindados por autores tales como Grover & Pea (2018), Brennan & Resnick (2012) y Weintrop et al (2016), y relacionarlos con los distintos conceptos y prácticas que ejercen los distintos programas de tipo software, aplicaciones móviles o experiencias desconectadas.

El entorno de desarrollo que se analizó es MIT App inventor, este entorno lo podemos encontrar en el siguiente enlace: https://appinventor.mit.edu/ , cuenta con un lenguaje de programación intuitivo, donde no necesariamente se requiere ser programador para crear una aplicación móvil. Este tipo de software cuenta con una interfaz similar a Scratch, pues promueve la programación por bloques para desarrollar diversos tipos de algoritmos o secuencias para la realización de una acción. Este software es utilizado para la realización de juegos o proyectos para incentivar la enseñanza y el aprendizaje en áreas como la matemática, ciencias, informática, ingles entre otras ramas de estudio.

Son varias las ventajas de uso en este tipo de entornos de desarrollo de programación, una de ellas es elaborar recursos informáticos para la solución de problemas, pero también respecto a la creatividad y estrategias de creación de contenido que se llevan a cabo para contribuir de manera general a otros usuarios que se encuentren interesados en construir otro tipo de proyectos a nivel empresarial o educativo.

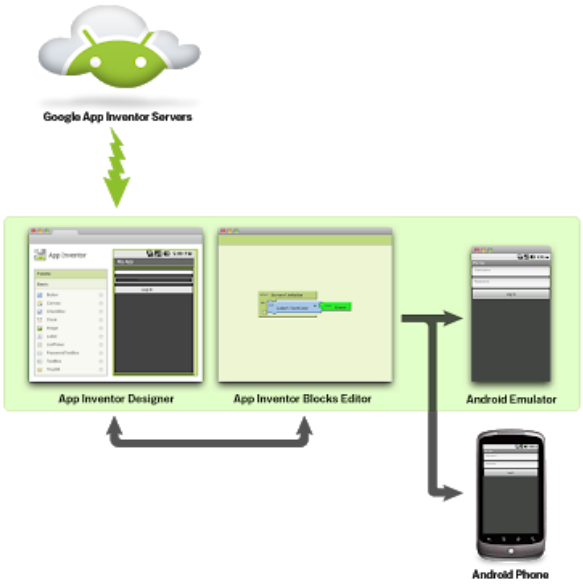
En la experiencia personal y formativa como docente del área de tecnología en los distintos escenarios educativos y alternativos, fue un factor preponderante la utilización de este tipo de herramientas, principalmente en ciclos de educación primaria para comprender temáticas como maquinas simples y compuestas, sistemas mecánicos, energías renovables y no renovables e incluso energías aplicadas en el desarrollo de la física entre otras temáticas, en los que tuve la oportunidad de realizar entornos, para fomentar el aprendizaje a través del juego. Los resultados obtenidos fueron significativos, debido a la alta interacción de los estudiantes y la resolución de problemas en entornos didácticos acordes a los objetivos generales del curso. A continuación se mostrara el análisis realizado para App Inventor y su relación con la educación en el pensamiento computacional.

### 3.1.2 Propósito

Programo Ergo Sum, (2021) define a App Inventor como: “es un entorno de desarrollo de software para la elaboración de aplicaciones destinadas al sistema operativo de Android. El lenguaje es gratuito y se puede acceder fácilmente de la web. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil”.

**Figura 3.1**

*Diagrama de funcionamiento de App Inventor.* (Creative Commons, 2012).




***Fuente:*** Capítulo 3. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

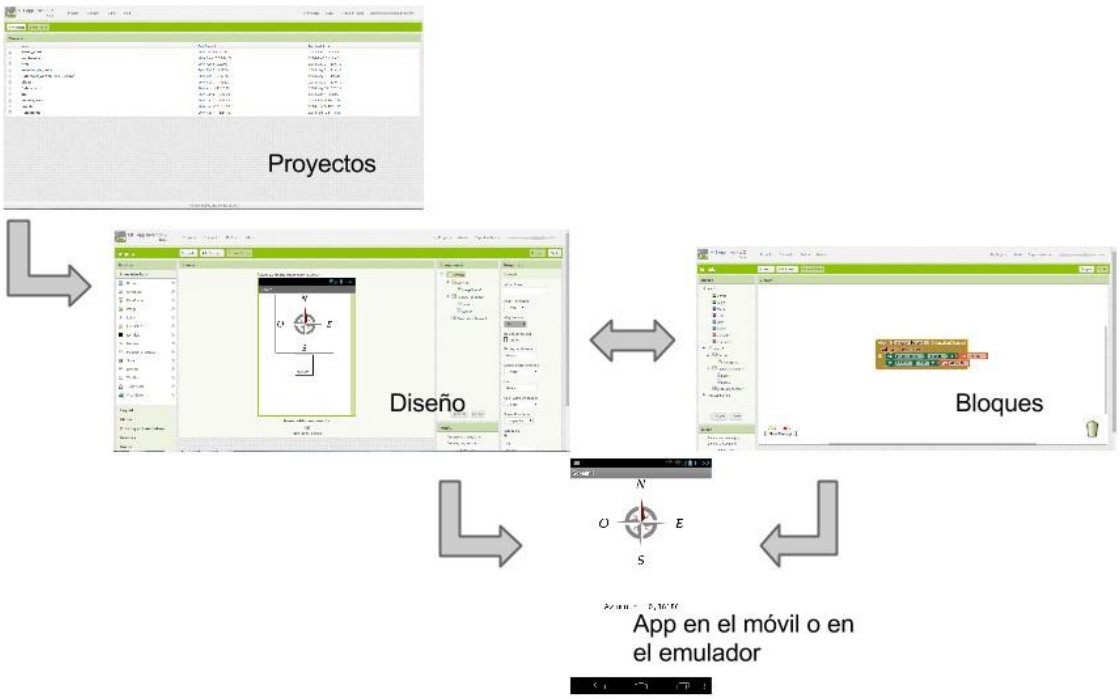
### 3.1.3 Público objetivo

Según Dédalo, (2013) está dirigido a personas que no están familiarizadas con la programación informática. Antes de salir al mercado se ha probado en diferentes centros educativos y la han utilizado desde niños de 12 años hasta licenciados universitarios sin nociones de programación. De manera visual y mediante un conjunto de herramientas básicas, el alumno puede crear una aplicación a través de un sistema de programación en bloques, así como sus propios videojuegos para el dispositivo móvil. Además ha tenido una notable aceptación debido

a que es gratis y que permite que todos los usuarios puedan compartir sus contenidos de forma libre y con mucha facilidad.

**Figura 3.1.2**

*Entorno de desarrollo App Inventor*. (App inventor en español, s.f.)



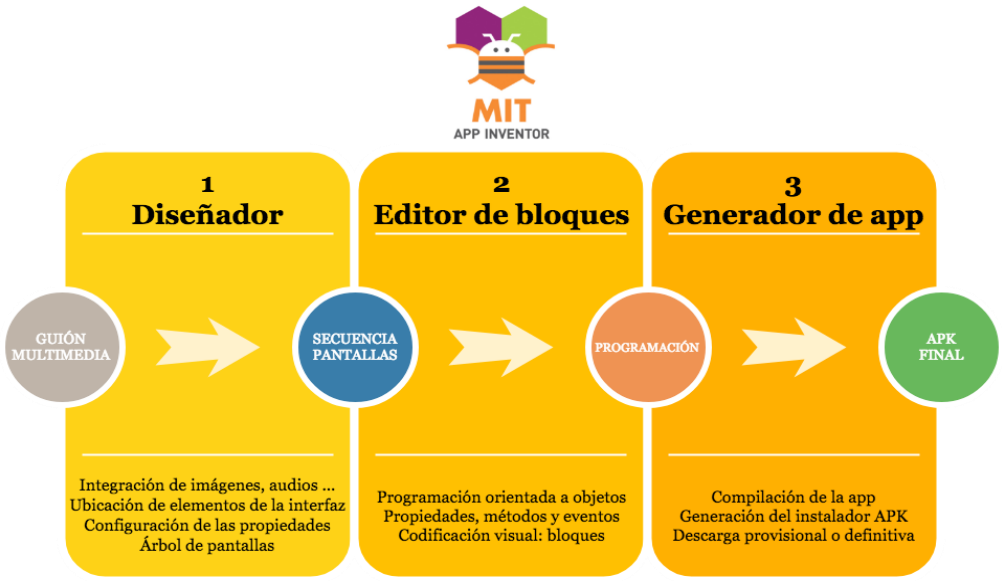

***Fuente:*** Capítulo 3. Visualización de panel App Inventor. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 3.1.4 Descripción del entorno

A continuación se observa 3 componentes sustanciales en la interfaz de App inventor, estos son: diseñador, editor de bloques y generador de app.

**Figura 3.1.3**

Proceso de creación de una app con MIT App Inventor (Prieto, s.f.)



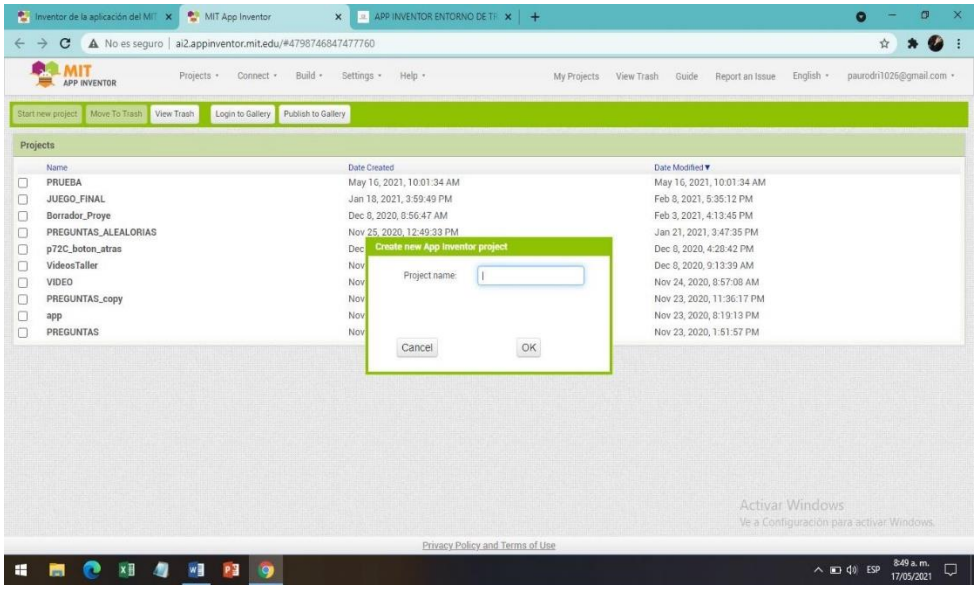

***Fuente:*** Capítulo 3. Visualización de interfaz App Inventor. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 3.1.5 Layout de trabajo

Si seleccionamos un proyecto o creamos uno nuevo, se cargará una nueva ventana

**Figura 3.1.4**

*Crear nueva ventana*




***Fuente:*** Capítulo 3.Elaboración propia.Nueva ventana App Inventor. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Encontraremos una serie de opciones que pasaremos a describir:

### 3.1.6 Screen 1

Este es el nombre que recibe por defecto la primera pantalla de nuestro proyecto. Resaltar que la primera pantalla no se puede borrar ya que va ligada a la creación del proyecto. Añadir que tampoco se le puede cambiar el nombre a esta primera pantalla, pero si a las sucesivas pantallas que creemos, Inventordeaplicaciones (2017).

**Figura 3.1.5**

*MIT App Inventor, componente ventana 1.*

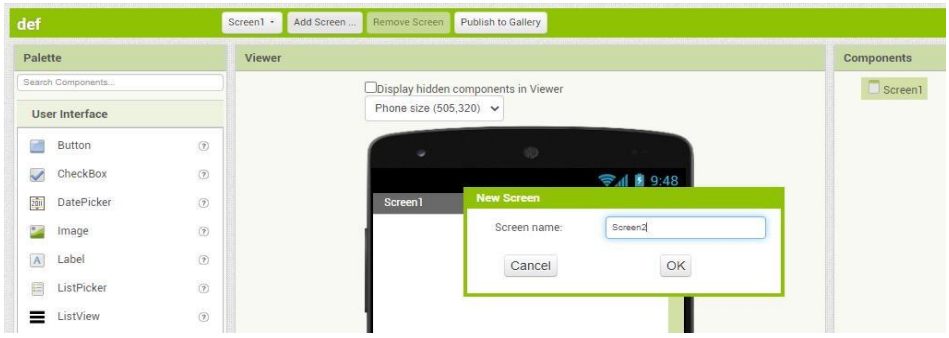

***Fuente:*** Capítulo 3.Elaboración propia. Componente ventana 1. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 3.1.7 Añadir ventana

Si pulsamos este botón, se nos abrirá una nueva ventana donde se nos pedirá el nombre que queremos darle a la nueva pantalla, este puede ser modificado después, y no debería de contener caracteres extraños como espacios y símbolos como el # o arroba. Inventordeaplicaciones (2017).

**Figura 3.1.6**

MIT App Inventor, añadir ventana.



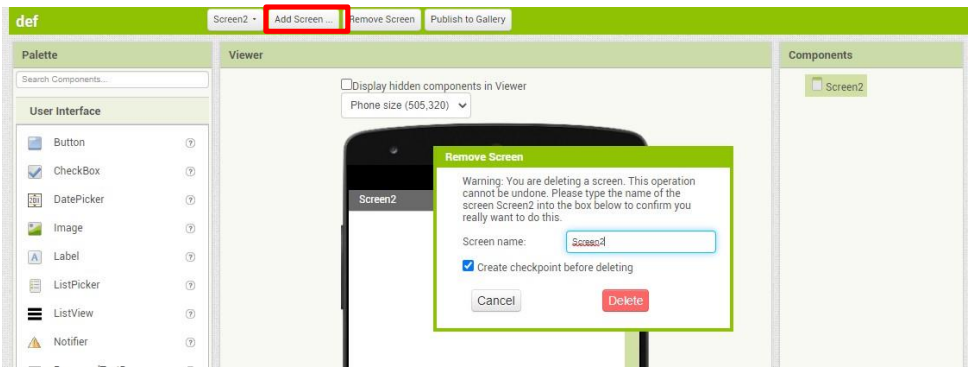

***Fuente:*** Capítulo 3.Elaboración propia. Añadir ventana. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 3.1.8 Eliminar ventana

Al pulsar se eliminará la ventana que esté seleccionada en ese momento, también se eliminará todos sus componentes y bloques que pudiera contener esta pantalla, Inventordeaplicaciones (2017).

**Figura 3.1.7**

*MIT App Inventor, eliminar ventana.*




***Fuente:*** Capítulo 3.Elaboración propia. Eliminar ventana. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Al construir las aplicaciones para Android trabajarás con dos herramientas: App Inventor Diseñador y App Inventor editor de bloques En el diseño de la aplicación construirás el Interfaz de Usuario, eligiendo y situando los elementos con los que interactuará el usuario y los componentes que utilizará la aplicación, App Inventor en español, (s.f.).

**Figura 3.1.8**

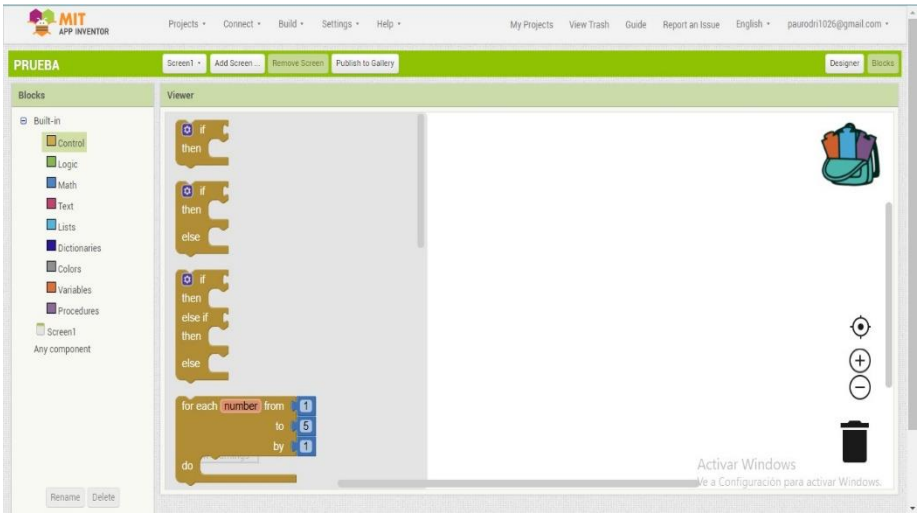
*MIT App Inventor, Componente Diseñador y bloques.*


***Fuente:*** Capítulo 3.Elaboración propia. Botonera de diseñador y bloques. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

El editor de bloques, donde irá escogiendo los bloques que te sean necesarios según la aplicación que proyectado realizar.

**Figura 3.1.9**

*MIT App Inventor, Interfaz de bloques.*

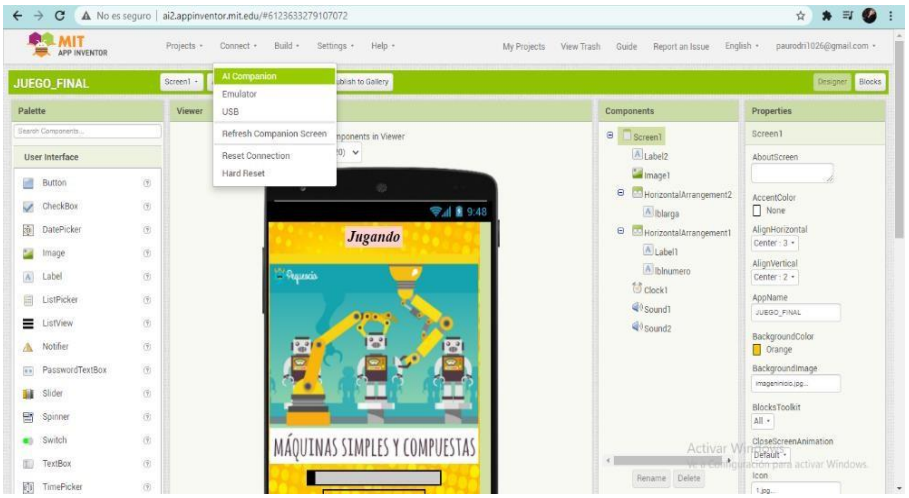



***Fuente:*** Capítulo 3.Elaboración propia. La interfaz de bloques dividida por variables de lógica. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

La aplicación aparece paso a paso de en la pantalla del teléfono a medida que vamos añadiendo elementos, así podemos ir probando a través de un emulador o el mismo programa generará un código QR para leerlo en un celular android, descargando previamente una aplicación en Play Store llamada MIT AI2 Companion reproduciendo, dándole funcionalidad a la aplicación.

**Figura 3.2**

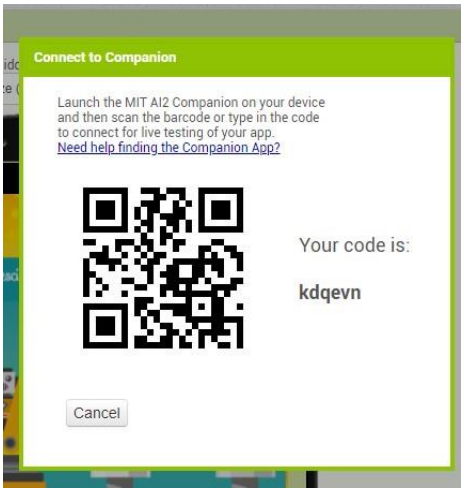
*MIT App Inventor, Reproducir aplicación móvil.*




***Fuente:*** Capítulo 3.Elaboración propia. Reproducir proyecto en emulador o aplicación móvil. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

**Figura 3.2.1**

*MIT App Inventor, código QR.*

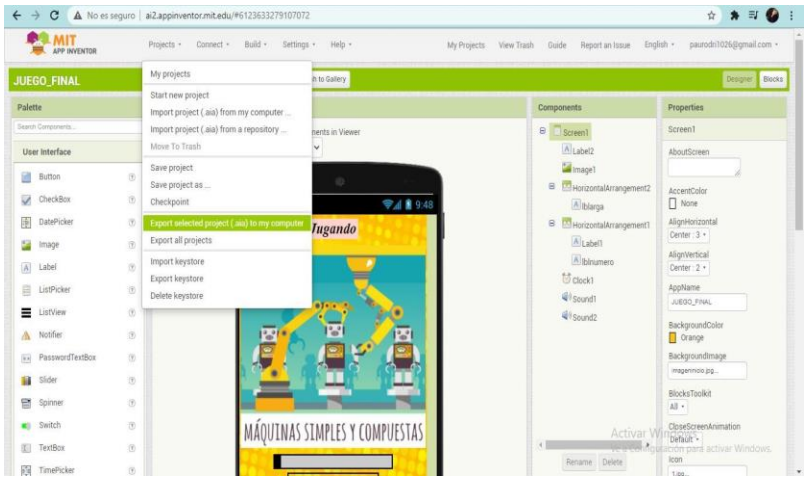



***Fuente:*** Capítulo 3.Elaboración propia. Imagen QR visualizador de juego en emulador o aplicación móvil. Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Cuando hemos acabado podemos importarla o exportarla en formato (.aia).

**Figura 3.2.2**

*MIT App Inventor, importar y exportar archivo.*

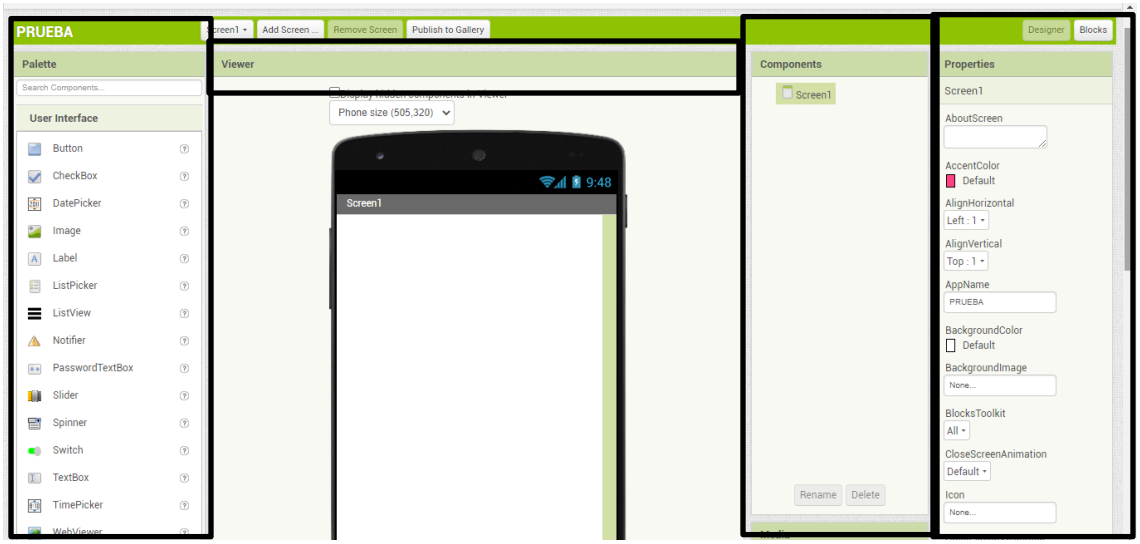



***Fuente:*** Capítulo 3.Elaboración propia, opción de exportar o importar archivo al software.Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 3.1.9 Diseño de interfaz

**Figura 3.2.3**

*MIT App Inventor, componentes y propiedades.*




***Fuente:*** Capítulo 3.Elaboración propia, componentes de creación App inventor .Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

El entorno App Inventor según el blog Creacion de apps educativas,(s.f.) se encuentra dividido en 4 paneles:

Diseñador: Se organiza en cuatro paneles, al pulsar este botón nos mostrará un conjunto de secciones en donde podremos diseñar el aspecto de nuestra aplicación.

Paleta: En la paleta podrá seleccionar los componentes que incluirá en su aplicación: En componentes visibles, que son todos los que el usuario pueda interactuar, como botones, selectores de flecha o campos de texto, y los componentes no visibles, como relojes temporizadores, sonidos o bases de datos.

Visor: simula la pantalla del móvil. En ella situará los componentes arrastrándolos desde la paleta.

Componentes: Presenta la lista de componentes y medios incluidos en la aplicación.

Propiedades: Presenta la lista de propiedades y sus valores actuales para el componente seleccionado en visualizador, por ejemplo en un botón podremos cambiar el texto fuente de la letra e incluso el color.

### 3.1.1.1 Editor de bloques

Para Ricoy, (2011) el editor de bloques tiene tres fichas en la esquina superior izquierda: Built-In (incorporados) y My Blocks (mis bloques). Los botones de debajo de cada ficha se amplían y se muestran los bloques cuando se hace clic. Los bloques Built-In son el conjunto estándar de bloques que están disponibles para cualquier aplicación que construya.

**Figura 3.2.4**

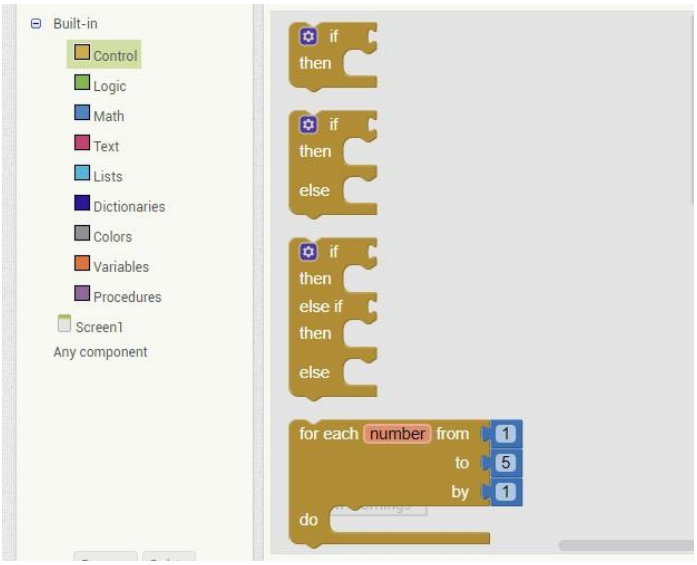
*MIT App Variables de editor de bloques.*


***Fuente:*** Capítulo 3.Elaboración propia, visualización de variables por bloques de colores.Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

Los bloques contienen bloques específicos que están vinculados al conjunto de componentes que has elegido para la aplicación. Dentro de bloques, ordenados por funciones y colores, encontramos distintos componentes de lógica de nuestro programa, desde funciones matemáticas, variables, funciones para crear bucles o incluso para crear colores.

**Figura 3.2.4**

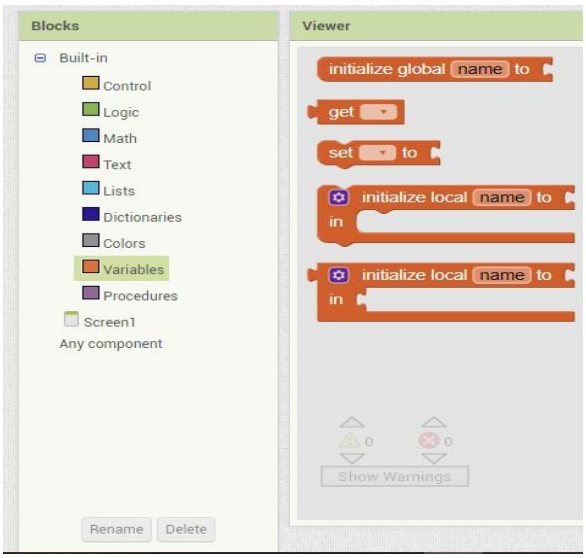
*MIT App Componente de control.*




***Fuente:*** Capítulo 3.Elaboración propia, la imagen muestra el componente de control .Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

**Figura 3.2.5**

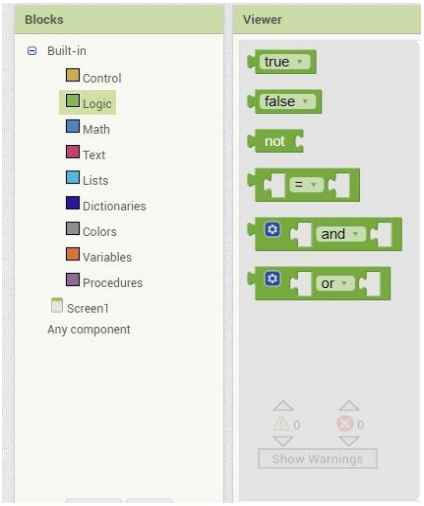
*MIT App Variables.*




***Fuente:*** Capítulo 3.Elaboración propia, la imagen muestra las variables de control .Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

**Figura 3.2.6**

*MIT App Lógica.*




***Fuente:*** Capítulo 3.Elaboración propia, la imagen muestra las variables de lógica.Análisis de aplicaciones y actividades para el desarrollo del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 3.2 Aportes del escenario al desarrollo del pensamiento computacional

### 3.3 Desde la perspectiva de pensamiento computacional de Grover y Pea

App inventor en conjunto a las ideas de (Grover & Pea, 2013) buscan generar procesos y prácticas de acuerdo a las siguientes características:

1. Formular problemas de manera que nos permita utilizar una computadora y otras herramientas digitales para ayudar a resolverlos.
2. Organizar y analizar datos de forma lógica; La construcción de proyectos educativos por medio de aplicaciones móviles, promueve el desarrollo cognitivo ideando estrategias para la construcción de una tarea específica. Además es un software que cuenta con una herramienta visual muy fácil de utilizar en la creación de entornos visuales como también encontramos variables de lógica que representan una serie de condicionales para iniciar una acción.
3. Representar datos a través de abstracciones como modelos y simulaciones por medio de la creación de bloques lógicos, diferenciando un proceso de otro que se va a ejecutar dentro del mismo entorno interactivo.
4. Automatizar soluciones a través del pensamiento algorítmico, refiriéndose a una serie de pasos ordenados, para hallar una estructura lógica de programación.
5. Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más eficiente y efectiva de pasos y recursos, dado por la construcción de dinámicas de jugabilidad.

6. Generalizar y transferir este proceso de resolución de problemas a una amplia variedad de problemas en distintos campos del saber.

### 3.4 Desde la perspectiva de pensamiento computacional de Brennan y Resnick

Brennan & Resnick , (2012) en su artículo “New frameworks for studying and assessing the development of computational thinking” contempla conceptos, prácticas y perspectivas que se desarrollan en la creación digital, ya que por medio de este tipo de interacciones se evidencian secuencias, condicionales, bucles, operadores gráficos y paralelismos que nos permite solucionar un programa computacionalmente.

Estos elementos ya vienen con bloques predefinidos y de acuerdo a la elección del usuario idealiza un esquema mental sobre los elementos a seleccionar, cuáles serían útiles buscar y aquellos que considera descartar en la depuración de la programación. Fomenta el trabajo colaborativo en los estudiantes.

### 3.5 Desde la perspectiva de pensamiento computacional de Weintrop

Por su parte Weintrop (2015), defiende el enfoque de incorporar el pensamiento computacional en contextos matemáticos y científicos. De ese modo, la aplicación App Inventor permite crear proyectos de todo tipo, en los tópicos del modelo STEM, potencializando otras áreas en el campo de las ciencias.

# Capítulo 4

## 4. Enfoques de evaluación del pensamiento computacional

Para Robles & Román, (2018) el pensamiento computacional cuenta con un modelo de 3 dimensiones, los cuales se considera que es fundamental implementar y evaluar, estos están mediados por los siguientes cuestionamientos:

- ¿Qué aprender?

Esta mediado por conceptos computacionales, tomando elementos como secuencias, bucles, eventos, paralelismos, condicionales u otros que se fundamentan en las diversas estructuras de programación.

- ¿Cómo lo aprende?

Está conformado por las prácticas computacionales, es decir el estudiante aprende a través de los siguientes componentes:

1. La experimentación e iteración para realizar procesos sencillos hasta los más complejos.
2. La depuración: Consiste en tomar programas que posean fallas e identificar el problema y solucionarlo.
3. La remezcla: En desarrollos digitales, se toman las líneas de código y las unen con proyectos de otras personas para complementar dichas estructuras.
4. La modularización: Conlleva a trabajar en equipo, donde cada persona una parte de un proyecto y posteriormente lo unen para conformar una sola estructura.

- ¿Para qué aprender?

Adquiere en el aprendiz perspectivas computacionales como:

1. Expresarse y crear proyectos tecnológicos
2. Conectarse con la comunidad con el objetivo de compartir experiencias de aprendizaje y construir en conjunto sistemas complejos.
3. Para interrogarse e informarse acerca del mundo digital.

## 4.1 Enfoques de evaluación del pensamiento computacional

Algunos investigadores como Román-González, Moreno-León y Robles (2019) han reconocido la necesidad de vincular criterios y estrategias de evaluación, que permitan determinar una valoración de conocimientos, habilidades cognitivas y desempeños sobre el pensamiento computacional en el aula. Es oportuno delimitar sobre las competencias en los estudiantes y todos aquellos que pueden ser adaptables hacia las diferentes disciplinas (genéricas o transversales) ofrecen más oportunidades de aplicación y de aprendizajes.Molina Patlán , Morales Martínez , & Valenzuela González, (2016).

Para Leal-Urueña, (2021) en el módulo 4 evaluación del pensamiento computacional, menciona cinco enfoques para evaluar esta competencia:

### 4.1.1 Enfoque diagnóstico

El propósito de la evaluación diagnostica se centra en medir las capacidades de la persona por medio de desempeños, este tipo de enfoque realiza una prueba inicial y final con el propósito

de analizar el proceso de aprendizaje inicial y compararlo con los avances de aprendizaje final. El test así mismo una herramienta de apoyo para docentes y psicólogos para identificar los puntos fuertes y áreas de mejora, y de acuerdo a estos dos aspectos realizar un plan de mejora en el refuerzo de las dificultades en el estudiante, Web del maestro (2020).

### 4.1.2 Enfoques de logro (sumativa)

La evaluación sumativa consta en realizar un balance de los resultados aprendidos en el estudiante, en relación a los conceptos computacionales que se lograron vivenciar en el campo social o pedagógico.Estos pueden ser útiles para reestructurar las estrategias de enseñanza y los alcances futuros que se pueden concebir en el aprendizaje.

Uno de los modelos jerárquicos para clasificar los diferentes objetivos de aprendizaje es la taxonomía de Bloom, que permite en el estudiante retener y comprender datos, relacionarlos entre si y sintetizarlo en un nivel básico hasta el de mayor complejidad.

Dicha taxonomía establece seis niveles con una gradualidad creciente. Cada nivel requiere que el alumno haya alcanzado los niveles anteriores. El equipo de Benjamin Bloom jerarquizaba el ámbito cognitivo de la siguiente forma: Conocimiento, comprensión, aplicación, análisis, síntesis y evaluación (Docentes al día, 2019)

### 4.1.3 Enfoques Formativo-Iterativos

El objetivo de la evaluación formativa consiste en realizar una retroalimentación de tareas, proyectos a nivel de programación, talleres entre otros insumos académicos realizados por el estudiante, en los que se evalúan conocimientos, actitudes y habilidades en el marco del pensamiento computacional.

El docente por medio de la observación y conocimientos previos, contempla criterios de evaluación conformados por fortalezas, debilidades y estilos de aprendizaje, en proporción al proceso de cada estudiante y determina planes de mejora respecto a las malas prácticas y dominios conceptuales.

Al respecto Carrillo, (2019) en el análisis de la evaluación formativa toma como referente autores como Pérez, Clemente y López (2009), al respecto se indica:

*"*Una mirada desde la evaluación formadora concebida como una estrategia que promueve la autorreflexión y el control sobre el propio aprendizaje que pretende desarrollar en los estudiantes competencias y/o estrategias de autodirección en el aprendizaje y generar proceso metacognitivos, no simplemente utilizar las técnicas citadas (autoevaluación, evaluación mutua y coevaluación) no se trata de explicar al estudiante las probables causas de sus errores y lo que puedan hacer para evitarlos, sino que sean ellos mismos los que tengan que explicárselos a sí mismos o a otros compañeros".

Otra de las estrategias para implementar este enfoque de evaluación, son las aplicaciones web de análisis automático que permiten analizar entornos de programación como Scratch, CodeMaster en tiempo real, para determinar una valoración de los conocimientos, referente a los pilares desarrollados en la lógica del pensamiento computacional.

### 4.1.4 Evaluación con minería de datos

"El objetivo general del proceso de minería de datos consiste en extraer información de un conjunto de datos y transformarla en una estructura comprensible para su uso posterior". (Vasquez espinosa, 2021)

Les permite a los educadores acceder digitalmente a los datos de los estudiantes en tiempo real, realizando una revisión minuciosa sobre la información que expone el proyecto, identificar errores y corregirlos para su posterior publicación, puesto que sería útil emplearlo como material de apoyo educativo tras su perfeccionamiento. También es un enfoque para comprobar los aprendizajes cognitivos en los estudiantes buscando patrones que sean precisos, comprensibles e interesantes.

### 4.1.5 Enfoque de transferencia de habilidades

En la transferencia de habilidades, el sujeto relaciona sus aprendizajes de pensamiento computacional con otras experiencias y contextos, ya sea en la cotidianidad como en el ámbito educativo. Estos conocimientos pueden estar fundamentados desde cualquier disciplina para aplicarlos en la solución de problemas.

El instituto nacional de tecnologías educativas y de formación del profesorado, promovió una iniciativa llamada *Bebras Contest Bebras* en el que se promueve la Informática y el pensamiento computacional entre los estudiantes de todas las edades. Los participantes generalmente son supervisados por maestros que pueden integrar el Bebras Contest en sus actividades de enseñanza. El desafío se realiza en escuelas que usan ordenadores o dispositivos móviles. Cabe destacar que estos conocimientos adquiridos a lo largo de la vida, permiten un mejor desempeño laboral y profesional al beneficio propio o en general al servicio de una comunidad, Intef, (2017).

## 4.2 Profundización en los enfoques de evaluación del pensamiento computacional

La actividad de profundización consiste en seleccionar un artículo de investigación sobre los distintos enfoques de evaluación del pensamiento computacional, realizando una lectura precisa del artículo para comprender esta visión y luego realizar una presentación por medio de 3 diapositivas sustentando los aspectos más relevantes de este enfoque.

### 4.2.1 Artículo seleccionado

Con base a los recursos alojados en el modulo 4 , el articulo de investigación seleccionado ha sido desarrollado por Román Gonzalez , Pérez González, & Jiménez Fernández, (2015) titulado "Test de Pensamiento Computacional: diseño y psicometría general" este está dirigido a una población de españoles entre 12 y 13 años, a partir de una variable de comportamiento psicométrico en el que se tuvo como referencia 400 personas durante su aplicación.

### 4.2.2 Resumen del artículo

Se sintetiza la importancia de dominar el lenguaje de programación 'código alfabetización' como habilidad esencial en la actual sociedad digital, en el que las personas tengan la capacidad de leer y escribir en el lenguaje de los ordenadores y maquinas. Varios países del mundo han concebido el PC como una habilidad que se debe fomentar en todas las disciplinas STEM (ciencia, tecnología, ingeniería y matemáticas).

Resnick (2009) concuerda que el pensamiento computacional es habilidad potencial para la creación de proyectos y erradicar la dependencia del consumo tecnológico. Por lo tanto la programación informática es una de las asignaturas que se han estado implementando en los currículos educativos en diversos países del mundo por medio del *coding.*

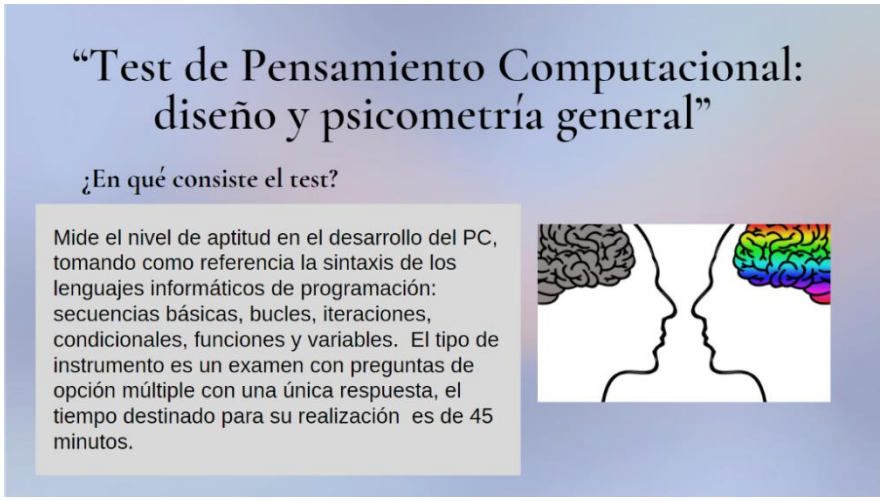
Actualmente no hay una definición exacta de lo que significa el PC o una precisión de cómo integrarlo en los currículos educativos, esto implicaría un vacío sobre cómo medir y evaluar esta competencia. Por consiguiente se han diseñado instrumentos de medida para verificar en qué grado los estudiantes han desarrollado los distintos tipos de habilidades.

### 4.2.3 ¿En qué consiste el test?

Valida el nivel de aptitud en el desarrollo del PC, tomando como referencia la sintaxis de los lenguajes informáticos de programación: secuencias básicas, bucles, iteraciones, condicionales, funciones y variables. El tipo de instrumento es un examen con preguntas de opción múltiple con una única respuesta, el tiempo destinado para su realización es de 45 minutos.

**Figura 4.1**

*Diapositiva 1 Artículo de investigación*




***Fuente:*** Capítulo 4 .Elaboración propia, especifica una idea conceptual del Test.Enfoques de evaluación del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

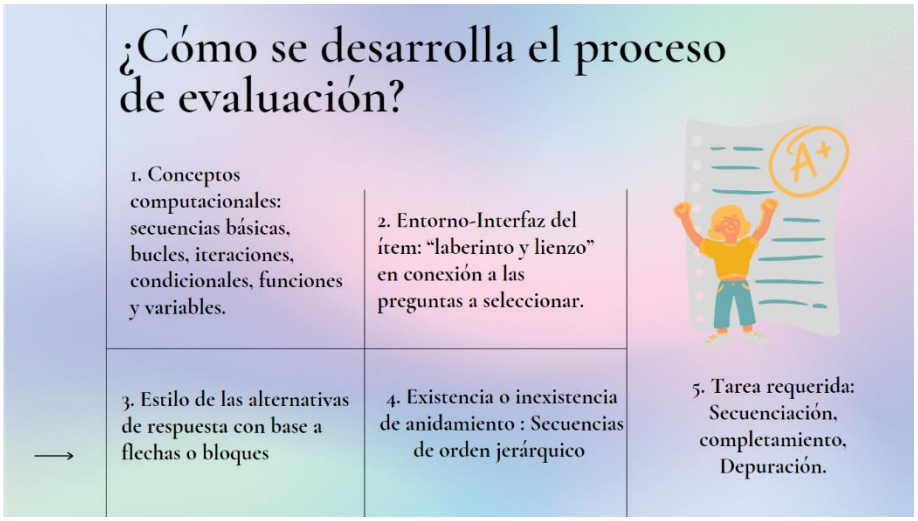
### 4.2.4 ¿Cómo se desarrolla el proceso de evaluación?

Se evalúa el test teniendo en cuenta 5 dimensiones:

1. Conceptos computacionales: secuencias básicas, bucles, iteraciones, condicionales, funciones y variables.
2. Entorno-Interfaz del ítem: "laberinto y lienzo" en conexión a las preguntas a seleccionar.
3. Estilo de las alternativas de respuesta con base a flechas o bloques
4. Existencia o inexistencia de anidamiento: Secuencias de orden jerárquico
5. Tarea requerida: Secuenciación, completamiento, Depuración.

**Figura 4.2**

*Diapositiva 2 Articulo de investigación.*



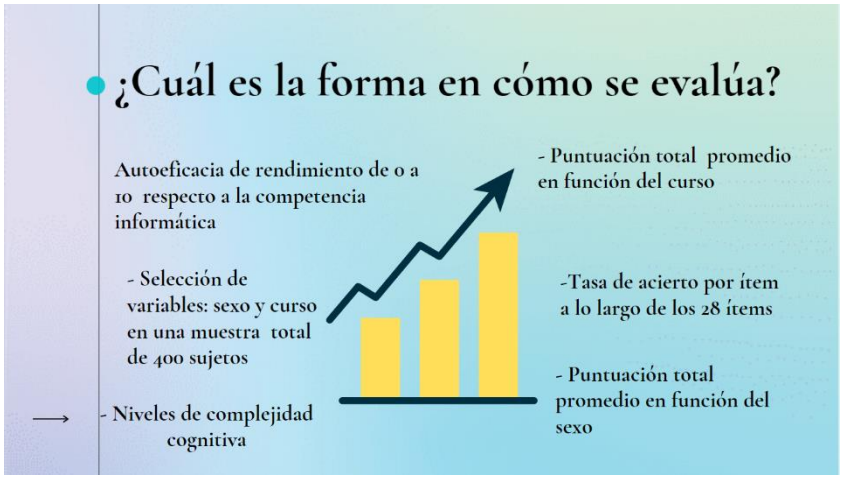

***Fuente:*** Capítulo 4 .Elaboración propia, especifica el proceso de evaluación del Test.Enfoques de evaluación del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

### 4.2.5 ¿Cuál es la forma en cómo se evalúa?

- Se evalúa de manera estadística teniendo en cuenta las siguientes variables:

**Figura 4.3**

*Diapositiva 3 Articulo de investigación.*




***Fuente:*** Capítulo 4 .Elaboración propia, especifica las variables de evaluación luego de la experimentación del Test.Enfoques de evaluación del pensamiento computacional. Asignatura Pensamiento computacional propuesta por la profesora Linda Leal-Urueña.

# 5. Conclusiones

La construcción de este informe permitió analizar la importancia para los futuros docentes de vincular esta competencia en diversos escenarios educativos, con el objetivo de replantear las formas de enseñanza tradicionales y construir nuevos esquemas de razonamiento para la resolución de problemas en las distintas áreas del conocimiento y situaciones de la vida cotidiana.

En este sentido, el pensamiento no solo se debe desarrollar en los profesionales de las ciencias de la computación, sino en personas que no poseen grandes conocimientos en los lenguajes de programación, puesto que permite desarrollar procesos cognitivos por medio de estructuras mentales para identificar un ejercicio complejo, analizar las posibles soluciones y finalmente hallar una solución general.

En ese orden de ideas, los aprendizajes fueron significativos puesto que se abordaron elementos tales como la fundamentación de conceptos, técnicas, marcos conceptuales y distintos modos de evaluación del pensamiento computacional. Luego de la revisión de estos contenidos, se realizó un análisis reflexivo de las temáticas resolviendo en detalle cada una de las lecciones y actividades, las cuales brindan una reconstrucción lógica de los procesos y los factores que intervinieron en su solución.

La fundamentación teórica fue esencial, puesto que permitió tener una mayor claridad de los conceptos y principios, para luego ponerlos en práctica por medio de actividades lúdicas de lógica y razonamiento. Estos procesos se resolvieron de manera descriptiva para generar una secuencia de pasos precisos y luego transformarlos en una estructura logarítmica, entendible y generalizada para que cualquier persona o maquina por medio de esta técnica lo logre leer fácilmente.

Se logró reconocer algunas técnicas algorítmicas de búsqueda ordenación aprendidos por medio de la danza, lo cual fue una estrategia creativa para comprender de una manera didáctica procesos complejos de programación, a través de una secuencia de pasos en los bailarines. Por otro lado, los diagramas de flujo en la informática, fueron útiles para la resolución de problemas matemáticos, llegando a la conclusión que una temática puede enseñarse desde cualquier disciplina, solo si la estrategia de enseñanza está debidamente estructurada.

Pudo reconocerse la importancia de la educación en el pensamiento computacional, respecto al uso de recursos informáticos como (software, simuladores, programas, actividades conectadas o desconectadas) útiles para fomentar las habilidades conceptuales, prácticos y de perspectiva en los distintos niveles de educación.

Uno de los entornos más propicios de interacción es App Inventor, siendo una de las herramientas de programación más acordes para la creación de proyectos, en su desarrollo se conciben aspectos como la abstracción, descomposición, reconocimiento de patrones y estructuras algorítmicas).

Finalmente, se logró profundizar sobre los distintos enfoques de evaluación para el pensamiento computacional en el aula, con el fin de medir los desempeños y aspectos de mejora en los estudiantes. Entre ellos, los instrumentos para la evaluación del pensamiento computacional, los cuales son fundamentales para realizar una valoración de los aprendizajes, fortaleciendo las formas de enseñanza y la alfabetización en las competencias para el siglo XXI.

## Referencias bibliográficas

App inventor en español. (s.f.). Obtenido de Primeros pasos: https://sites.google.com/site/appinventormegusta/primeros-pasos

App Inventor en español. (s.f.). Obtenido de Primeros pasos: https://sites.google.com/site/appinventormegusta/primeros-pasos

Brennan, K., & Resnick, M. (2012). Using artifact-based interviews to study the development of computational thinking in interactive media design. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

Carrillo, L. S. (3 de Abril de 2019). La evaluación formativa ¿Un concepto en algunos casos difuso e impreciso o una práctica en el aula? Obtenido de: https://www.magisterio.com.co/articulo/la-evaluacion-formativa-un-concepto-en-algunos-casos-difuso-e-impreciso-o-una-practica-en

Code Intef. (20 de 05 de 2020). El marco de Brennan-Resnick: I. Conceptos. Obtenido de: http://code.intef.es/marco-de-brennan-resnick-conceptos/

Creacion de apps educativas sin saber programacion. (s.f.). Entorno de trabajo. Obtenido de: https://sites.google.com/site/appssinsaberprogramar/contenidos/appinventor/interface

Creative Commons, A.-S. (2012). MIT App Inventor. Obtenido de Imagen: http://appinventor.mit.edu/explore/content/what-app-inventor.html

David Weintrop, E. B. (8 de 10 de 2015). Definición del pensamiento computacional para las aulas de matemáticas y ciencias. Obtenido de: https://link.springer.com/article/10.1007/s10956-015-9581-5

Dédalo fundacion. (10 de Diciembre de 2013). App Inventor, crea tus propias 'Apps' para Android. Obtenido de: https://www.fundaciondedalo.org/Ultimas-Noticias/app-inventor-crea-tus-propias-aplicaciones-para-android.html

Docentes al día. (27 de Octubre de 2019). Qué es la Taxonomía de Bloom y para qué sirve? Obtenido de: https://docentesaldia.com/2019/10/27/que-es-la-taxonomia-de-bloom-y-para-que-sirve/

Edacom tecnologia educativa. (25 de febrero de 2020). Pensamiento computacional: Una necesidad en la educación. Obtenido de: https://blog.edacom.mx/pensamiento-computacional-necesario-educacion

Román, M., Pérez, J., & Jiménez, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general [Computational Thinking Test: design & general psychometry]. 10.13140/RG.2.1.3056.5521.

Grover, S., & Pea, R. (2013). Pensamiento computacional en K-12: una revisión del estado del campo. Obtenido de Educational Researcher, 42 (1), 38–43.

Intef. (25 de 10 de 2017). Participa en Bebras Contest. Obtenido de: https://intef.es/Noticias/bebras-contest/

Inventordeaplicaciones. (28 de Noviembre de 2017). App Inventor entorno de trabajo. Obtenido de: https://inventordeaplicaciones.es/app-inventor-desde-0/app-inventor-entorno-de-trabajo/

Leal-Urueña, L. (2021a). Marcos conceptuales y escenarios pedagógicos y tecnológicos - Parte 1. Obtenido de: https://upnvirtual.pedagogica.edu.co/mod/url/view.php?id=24318

Leal-Urueña, L. (2021b). Actividad 1. Poniendo en juego tu comprensión del pensamiento computacional. Obtenido de:

https://upnvirtual.pedagogica.edu.co/mod/lesson/view.php?id=12007&pageid=59&startlastseen=no

Leal-Urueña, L. (julio de 2020). Fundamentos de pensamiento computacional. Obtenido de: https://upnvirtual.pedagogica.edu.co/mod/resource/view.php?id=12005

Leal-Urueña, L. (2021c). Presentación del seminario electivo de pensamiento computacional. Obtenido de: https://upnvirtual.pedagogica.edu.co/mod/url/view.php?id=13919

Leal-Urueña, L. (2021). Lección 2. Introducción a las técnicas algorítmicas. Obtenido de: https://upnvirtual.pedagogica.edu.co/mod/lesson/view.php?id=12008&pageid=63&startlastseen=no

Leal-Urueña, L. (10 de Junio de 2021). Evaluación del pensamiento computacional. Obtenido de: https://upnvirtual.pedagogica.edu.co/mod/page/view.php?id=12010

Molina Patlán, C., Morales Martínez, G. P., & Valenzuela González, J. R. (1 de Enero de 2016). Competencia transversal pensamiento crítico: Su caracterización en estudiantes de una secundaria de México. Obtenido de https://www.scielo.sa.cr/pdf/ree/v20n1/1409-4258-ree-20-01-00237.pdf

Net- Learning. (s.f.). Pensamiento computacional: por qué incluirlo en el proceso de aprendizaje. Obtenido de: https://www.net-learning.com.ar/blog/herramientas/pensamiento-computacional-por-que-incluirlo-en-el-proceso-de-aprendizaje.html

Patiño Avendaño, B., & Charry, O. G. (Junio de 2013). LA ENSEÑANZA DE LA TEORÍA DE GRAFOS COMO. Obtenido de https://repository.usergioarboleda.edu.co/bitstream/handle/11232/844/La%20ense%C3%B1anza%20de%20la%20teor%C3%ADa%20de%20grafos%20como%20estrategia.%20procesos%20de%20matematizaci%C3%B3n.pdf?sequence=2&isAllowed=y

Prieto, F. P. (s.f.). Creando aplicaciones para móviles Android con MIT App Inventor. Obtenido de (Imagen): https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

Programo Ergo Sum. (2021). Curso de introducción a la programación de apps con AppInventor: Obtenido de: https://www.programoergosum.com/cursos-online/appinventor/27-curso-de-programacion-con-app-inventor/primeros-pasos

Proyecto pensamiento computacional. (s.f Pensamiento computacional. Obtenido de: https://unipe.educar.gob.ar/unipe/seccion/7/unipe

Ricoy Riego, A. (26 de Febrero de 2011). App inventor en español. Obtenido: https://sites.google.com/site/appinventormegusta/conceptos

Robles, G., & Román, M. (2018). Cursos de verano MECD-UIMP 2018. Pensamiento computacional en infantil y primaria. Obtenido de: https://youtu.be/m1ZZWkQAjl8

Sánchez-Vera, M. d. (Diciembre de 2019). Realia. El pensamiento computacional en contextos educativos: una aproximación desde la tecnología educativa.

Universidad Militar Nueva Granada. (s.f.). Algoritmos de ordenamiento y búsqueda. Obtenido de: http://virtual.umng.edu.co/distancia/ecosistema/odin/odin_desktop.php?path=Li4vb3Zhcy9pbmdlbmllcmlhX2luZm9ybWF0aWNhL2VzdHJ1Y3R1cmFfZGVfZGF0b3MvdW5pZGFkXzEv#slide_1

Vásquez espinosa, P. G. (13 de junio de 2021). Wikipedia. Obtenido de Analítica y Minería de Datos: https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos

Vilchez Degregori, J. C. (2017). Fundamentos de programación. Obtenido de Diseñar algoritmo de secuencia: https://sites.google.com/site/portafoliosenati/fundamentos-de-programacion/3-disenar-algoritmo-secuencia

Web del maestro cmf. (15 de Marzo de 2020). ¿Qué es la evaluación diagnóstica y para qué sirve? Obtenido de: https://webdelmaestrocmf.com/portal/que-es-la-evaluacion-diagnostica-y-para-que-sirve/

Wing, J. M. (2006 ). How I learned code. Obtenido de Jeannette M. Wing y el Pensamiento Computacional. Obtenido de https://howilearnedcode.com/2016/10/el-pensamiento-computacional-jeannette-m-wing/#:~:text=Wing%20(2006%2C%202010)%20defini%C3%B3,agente%20de%20procesamiento%20de%20informaci%C3%B3n.%E2%80%9D