

**Enseñanza de programación en Python para estudiantes sordos mediante un ambiente virtual de aprendizaje bilingüe (LSC- español)**

Gloria Andrea Caro Betancourt

Universidad Pedagógica Nacional  
Facultad de Ciencia y Tecnología  
Licenciatura en Diseño Tecnológico  
Bogotá, Colombia

2026

**Enseñanza de programación en Python para estudiantes sordos mediante un ambiente virtual de aprendizaje bilingüe (LSC- español)**

Gloria Andrea Caro Betancourt

Directora

Linda Alejandra Leal Urueña

Trabajo de grado para optar por el título de Licenciada en Diseño Tecnológico

Universidad Pedagógica Nacional

Facultad de Ciencia y Tecnología

Licenciatura en Diseño Tecnológico

Bogotá, Colombia

2026

**Dedicatoria.**

A mi esposo, César, por acompañarme en este proceso, por su apoyo incondicional y por creer en este proyecto desde el inicio. Gracias por aportar tus conocimientos, tu tiempo y tu paciencia para ayudarme a convertir esta idea en una realidad. A mi hijo, Filiph, por acompañarme durante mi carrera, por compartir conmigo tantos momentos y por permitirme disfrutar de la vida desde su alegría y curiosidad..

A mi mamá, Nury. Sé que desde el cielo hizo todo lo posible para que lograra ingresar a la Universidad Pedagógica Nacional, porque este era el último intento que haría para estudiar esta carrera, y ella se encargó de que esta vez sí fuera posible.

A mi papá, porque, aunque cada vez que nos vemos me pregunta qué fue lo que estudié y no logra recordarlo, su presencia es muy importante.

A mis hermanos, Diana, Claudia, Paola y Alejo, y a mis sobrinos, Alex, Mariana, Salomón, Gabriela, María José y Samantha, quienes me apoyaron cuando tuve dudas y me permitieron comprender mejor la educación a través de los trabajos, encuestas e investigaciones que realizamos juntos.

A Nakiry, Vaky y Sunny, por su compañía fiel durante las largas jornadas de trabajo.

Y a todas las personas que me acompañaron y creyeron en mí, gracias por ser parte de este proceso.

## **Agradecimientos**

Llegar a este momento me llena de una alegría profunda porque este título representa años de esfuerzo y de no rendirme ante los obstáculos.

Primero que todo, le agradezco a mi familia por estar ahí siempre, por apoyarme y no dejar que me rindiera cuando las cosas se ponían difíciles. Sin ustedes, no habría llegado hasta aquí.

Un agradecimiento muy especial para la profesora Linda Alejandra Leal. Fue una asesora siempre amable y con una paciencia increíble. Sus correcciones y consejos me dieron la seguridad que necesitaba para creer en mi trabajo y avanzar con claridad.

A la Universidad Pedagógica Nacional y a la Licenciatura en Diseño Tecnológico.

Agradezco a los profesores que me enseñaron que la educación es para todos y que la tecnología es un camino para lograrlo. Gracias a la universidad por abrirme sus puertas y mostrarme un mundo de luchas y esfuerzos, donde aprendí que hay personas que trabajan día a día para enseñarnos que siempre hay mucho más por aprender y transformar.

Al Colegio Isabel Segunda, por abrirme sus puertas y permitirme conocer de cerca las realidades de la educación inclusiva; esta experiencia me hizo ver que la educación puede generar oportunidades por igual.

Finalmente, agradezco a la vida por darme la oportunidad de estudiar en la universidad pública. Gracias por las lecciones y por la oportunidad de aprender y conocer.

## Resumen

El presente estudio tuvo como objetivo diseñar, implementar y evaluar un ambiente virtual de aprendizaje accesible para la enseñanza de conceptos básicos de programación en Python, dirigido a estudiantes sordos. La propuesta integró estrategias pedagógicas de comprensión, visualización y seguimiento de instrucciones, apoyadas en recursos visuales y Lengua de Señas Colombiana (LSC). La investigación se desarrolló bajo un enfoque mixto con la participación de seis estudiantes de educación media, a quienes se les aplicó un pretest y un postest para evaluar su desempeño. Los resultados evidenciaron una mejora moderada en el rendimiento académico, así como diferencias en la efectividad de las estrategias implementadas. La visualización favoreció respuestas rápidas y precisas en conceptos concretos como tipos de datos, mientras que la comprensión mediada por LSC facilitó la interpretación inicial de los enunciados. Sin embargo, se identificaron dificultades en la aplicación de estructuras condicionales, especialmente en situaciones que requerían autonomía. En conclusión, el aprendizaje de la programación en estudiantes sordos requiere la integración de múltiples estrategias pedagógicas y recursos accesibles, así como procesos de práctica más prolongados para lograr una apropiación completa del conocimiento.

*Palabras clave:* educación inclusiva, programación, estudiantes sordos, accesibilidad, ambientes virtuales de aprendizaje, Python.

## Contenido

Introducción.....	12
1. Formulación del problema.....	14
1.1 Justificación.....	14
1.2 El problema de investigación.....	17
1.3 Objetivos.....	21
1.3.1 <i>Objetivo general</i> .....	21
1.3.2 <i>Objetivos específicos</i> .....	21
2. Estado del arte y Marco teórico.....	23
2.1 Educación inclusiva y accesibilidad en población sorda.....	23
2.2 Pensamiento computacional y enseñanza de programación.....	24
2.3 Estrategias de enseñanza-aprendizaje efectivas para población sorda.....	25
2.3.1 <i>Comprensión</i> .....	26
2.3.2 <i>Visualización</i> .....	26
2.3.3 <i>Seguimiento de instrucciones</i> .....	27
2.3.4 <i>Involucrar a un observador par</i> .....	28
2.3.5 <i>Visualización del proceso de resolución de problemas</i> .....	29
2.3.6 <i>Atención a las instrucciones del profesor durante la codificación</i> .....	30
2.4 Ambientes virtuales de aprendizaje.....	31
3. Metodología.....	33
3.1 Enfoque metodológico.....	33
3.2 Descripción de la intervención.....	34
3.2.1 <i>Elección de estrategias</i> .....	35
3.2.2 <i>Diseño pedagógico</i> .....	35
3.2.3 <i>Programación del ambiente virtual de aprendizaje</i> .....	35

3.2.4. Prueba piloto.....	36
3.2.5. Gestión administrativa y consentimientos.....	36
3.2.6. Detalle de las sesiones de campo.....	37
3.3 Muestra del estudio.....	37
3.4 Instrumentos de recolección de información.....	40
3.4.1 Prueba diagnóstica y de salida (pretest y postest).....	40
3.4.2 Diario de campo y observación directa.....	41
3.4.3 Registro de métricas del sistema.....	42
3.5 Análisis de la información.....	42
4. Diseño del entorno de aprendizaje.....	44
4.1. Estructura de contenidos y estrategias de mediación.....	44
<i>Tema 1. Variables.....</i>	<i>44</i>
<i>A. Estrategia de comprensión: Variables.....</i>	<i>45</i>
<i>B. Estrategia de visualización: Variables.....</i>	<i>47</i>
<i>C. Contenido técnico (Python).....</i>	<i>48</i>
<i>D. Estrategia de seguimiento de instrucciones: Variables.....</i>	<i>51</i>
<i>Tema 2. Tipos de datos.....</i>	<i>53</i>
<i>A. Estrategia de comprensión: Tipos de datos.....</i>	<i>55</i>
<i>B. Estrategia de visualización: Tipos de datos.....</i>	<i>57</i>
<i>C. Estrategia de seguimiento de instrucciones: Tipos de datos.....</i>	<i>59</i>
<i>Tema 3. Condicionales.....</i>	<i>61</i>
<i>A. Estrategia de comprensión: Condicionales.....</i>	<i>62</i>
<i>B. Estrategia de visualización: Condicionales.....</i>	<i>64</i>
<i>C. Estrategia de seguimiento de instrucciones: Condicionales.....</i>	<i>66</i>
4.2. Arquitectura y desarrollo del sistema.....	68

5. Resultados.....	71
5.1 Información sociodemográfica de la población.....	71
5.2. Resultados generales en los conceptos de programación.....	72
5.3. Resultados de la estrategia de comprensión.....	77
5.4 Resultados de la estrategia de visualización.....	79
5.4.1 <i>Análisis de número de intentos en las estrategias de comprensión y visualización.....</i>	<i>81</i>
5.5 Resultados de la estrategia de seguimiento de instrucciones.....	83
6. Conclusiones.....	91
6.1 Limitaciones y recomendaciones para futuras investigaciones.....	95
Referencias.....	96
ANEXOS.....	100

### Índice de tablas

Tabla 1. Perfil sociodemográfico de los participantes .....	39
Tabla 2. Estructura del pretest y el postest.....	41
Tabla 3. Relación de participantes para el análisis de desempeño .....	72
Tabla 4. Resultados de la prueba diagnóstica (Pretest) y prueba final (Postest) .....	73
Tabla 5. Número de intentos por estrategia y unidad temática .....	81

### Índice Figuras.

Figura 1. Interfaz de la estrategia de comprensión concepto variables.....	46
Figura 2. Ejercicios de la estrategia de comprensión variables.....	47
Figura 3. Sentencia Print.....	48
Figura 4. Interfaz de la estrategia de visualización concepto variables.....	49
Figura 5. Código Python de cómo se ven las variables y la LSC al final.....	50
Figura 6. Ejercicios de la estrategia de visualización, variables.....	51
Figura 7. Interfaz estrategia seguimiento de instrucciones para el concepto de variables...	52
Figura 8. Ejercicio seguimiento de instrucciones para el concepto de variables.....	53
Figura 9. Explicación tipos de datos con numero entero.....	54
Figura 10. Explicación tipos de datos con números decimales.....	54
Figura 11. Explicación tipos de datos con valores Booleanos.....	55
Figura 12. Interfaz estrategia de comprensión concepto tipos de datos.....	56
Figura 13. Interfaz estrategia de comprensión ejercicio de tipos de datos.....	57
Figura 14. Explicación de tipos de datos mediante código Python y LSC.....	58
Figura 15. Ejercicio tipos de datos usando decimales.....	59
Figura 16. Interfaz de la estrategia de seguimiento de instrucciones, ejemplo Tipos de datos.....	60
Figura 17. Ejercicio de seguimiento de instrucciones con tipos de datos.....	61
Figura 18. Diagramas con apoyo de imágenes sobre la estructura de condicionales.....	62
Figura 19. Estrategia de comprensión, estructura condicional para observar la toma de	63

decisiones.....	
Figura 20. Estructura condicional con LSC, imágenes y texto para la toma de decisiones.	64
Figura 21. Interfaz estrategia de visualización, conceptos condicionales.....	65
Figura 22. Estrategia de visualización, ejercicio toma de decisiones con condicionales.....	66
Figura 23. Ejemplo de la estrategia de seguimiento de instrucciones.....	67
Figura 24. Ejercicio de condicionales con la estrategia seguimiento de instrucciones.....	68
Figura 25. Primera pregunta que se encuentra en el pretest y el posttest.....	73
Figura 26. Grafica que muestra una comparativa entre la prueba inicial y la final.....	76
Figura 27. Ejemplo de mediación (Sintaxis Python - Emoji - Señal LSC) .....	77
Figura 28. Ejemplo de código (Sintaxis Python - Emoji - Señal LSC) .....	80
Figura 29. Ejecución del reto final usuario 20260213105359-7.....	84
Figura 30. Ejecución del reto final usuario 20260213105914-1.....	85
Figura 31. Ejecución del reto final usuario 20260218104318-4.....	86
Figura 32. Ejecución del reto final usuario 20260218104307-2.....	87
Figura 33. Ejecución del reto final usuario 20260213105953-0.....	88
Figura 34. Ejecución del reto final usuario 20260218112145-5.....	89

## **Introducción**

La presente investigación surge de la necesidad de cerrar las brechas de acceso al conocimiento tecnológico para la comunidad sorda en Colombia, específicamente en el área de la programación de computadores. A pesar de los avances en inclusión educativa, la enseñanza de lenguajes de programación como Python en estudiantes con limitaciones auditivas ha estado limitada por barreras lingüísticas y la falta de recursos diseñados desde una perspectiva bilingüe (Lengua de Señas Colombiana – LSC, y español escrito).

El propósito central de este trabajo fue diseñar, implementar y evaluar un ambiente virtual de aprendizaje que, mediado por estrategias pedagógicas de comprensión, visualización y seguimiento de instrucciones específicas para estudiantes sordos, facilitara la asimilación de conceptos computacionales básicos en una muestra de estudiantes del Colegio Isabel Segunda, de la Localidad de Kennedy, en Bogotá. Esta intervención no solo buscó la enseñanza de conceptos computacionales básicos, sino también la validación de metodologías pedagógicas que reconocen la visualidad como el eje principal del aprendizaje en la población sorda.

A lo largo de este documento, el lector encontrará el desarrollo de esta propuesta, organizada en capítulos que detallan el problema de investigación, el estado del arte, su sustento teórico, la metodología de investigación, los pormenores del diseño del ambiente de aprendizaje

y los resultados obtenidos tras la intervención, donde se analizan las dinámicas de aprendizaje en cada una de las estrategias implementadas a través del entorno virtual de aprendizaje.

Este trabajo busca aportar al campo de la formación de profesores que desarrolla la Licenciatura en Diseño Tecnológico al proponer el uso de tecnologías educativas accesibles, el diseño de ambientes virtuales de aprendizaje inclusivos y el fortalecimiento del rol de diseñador tecnológico mediante la creación de soluciones pedagógicas que respondan a la diversidad de contextos educativos.

## **1. Formulación del problema**

### **1.1 Justificación**

La inclusión de la comunidad sorda en el sistema educativo continúa siendo un desafío para los docentes y las instituciones educativas. Los estudios revisados muestran brechas entre las expectativas de logros académicos y los resultados alcanzados por los estudiantes sordos, especialmente en contextos donde predominan metodologías centradas en el lenguaje oral y escrito (Mousley y Kelly, 1998; Celemín-Mora y Flórez-Romero, 2018). Aunque el Instituto Nacional para Sordos (INSOR, 2020) promueve la enseñanza de la Lengua de Señas Colombiana (LSC) como lengua principal y del español escrito como lengua secundaria, en la práctica este enfoque no se aplica de forma consistente dentro de las instituciones educativas.

Esto se refleja en el bajo desempeño de los estudiantes sordos en pruebas como Saber 11, en las que no se realizan adecuaciones pensadas para esta población (Celemín-Mora & Flórez-Romero, 2018). En el contexto colombiano, estos mismos autores analizaron las dificultades que enfrentan los estudiantes sordos en las pruebas Saber 11, cruciales para su acceso a la educación superior. Los autores observaron que las pruebas están diseñadas principalmente para estudiantes oyentes y no contienen adecuaciones suficientes o significativas para la población sorda, lo que resulta en un rendimiento inferior en comparación con sus pares oyentes. Aunque el ICFES ha implementado lineamientos y algunas adecuaciones orientadas a la inclusión de la población

sorda en las pruebas Saber 11 (ICFES, 2021), aún persisten dificultades relacionadas con el acceso lingüístico y la comprensión de los contenidos evaluados. Esta diferencia muestra la necesidad de continuar fortaleciendo estrategias y recursos que permitan garantizar procesos educativos y evaluativos más accesibles y equitativos para esta población.

A nivel pedagógico, los estudiantes sordos suelen tener menos acceso a explicaciones orales, discusiones de clase y materiales educativos. A lo anterior se suman las limitaciones en el acceso a la lectura y escritura en el idioma español, que en muchas ocasiones es su segunda lengua, lo que genera barreras adicionales para la interpretación de enunciados matemáticos o técnicos. Adicionalmente, la escasez de materiales visuales adaptados y de apoyos lingüísticos, como intérpretes especializados en cada contexto académico, afecta directamente su proceso de aprendizaje, especialmente en áreas donde predominan las explicaciones abstractas, extensas o basadas en el lenguaje escrito, como la programación de computadores.

Desde la educación inclusiva, el Diseño Universal para el Aprendizaje (DUA) plantea que es importante ofrecer distintas formas de presentar la información, participar y demostrar lo aprendido, para que todos los estudiantes puedan acceder al conocimiento sin importar sus condiciones (CAST, 2021).

La importancia de este proyecto radica en la necesidad de generar propuestas educativas inclusivas en el área de la programación, donde aún existen barreras de acceso para estudiantes sordos debido a la dependencia del lenguaje escrito y a que estos conceptos no son tan fáciles de entender al inicio. Por esta razón, el diseño del ambiente virtual no solo busca apoyar el aprendizaje de contenidos específicos, sino también aportar al fortalecimiento de prácticas pedagógicas más accesibles dentro de la educación media.

Desde el campo de la Licenciatura en Diseño Tecnológico, este trabajo aporta en tres dimensiones. En primer lugar, a nivel pedagógico, se propone una forma de enseñanza de la programación en Python para estudiantes sordos mediante apoyos visuales y recursos en Lengua de Señas Colombiana, reduciendo la dependencia del español escrito y fortaleciendo la comprensión de conceptos básicos de programación. En segundo lugar, a nivel tecnológico, el proyecto reflexiona sobre el diseño de ambientes virtuales de aprendizaje accesibles e inclusivos, que integren recursos digitales adaptados a diferentes necesidades comunicativas. Finalmente, a nivel social y metodológico, esta investigación aporta elementos para comprender cómo las estrategias pedagógicas y el uso de la tecnología pueden adaptarse para responder a contextos educativos diversos, promoviendo procesos de aprendizaje más equitativos.

## 1.2 El problema de investigación

A partir de la revisión de literatura realizada, el trabajo de Mousley y Kelly (1998), ofrece una perspectiva crítica sobre los métodos de enseñanza de matemáticas a estudiantes sordos, argumentando que las estrategias convencionales no siempre se adaptan a las necesidades específicas de estos estudiantes. Los enfoques tradicionales en la resolución de problemas a menudo ignoran las particularidades lingüísticas y culturales de estos estudiantes, lo que puede limitar su capacidad para desarrollar habilidades en esta área.

Mousley y Kelly (1998) sugieren que la implementación de estrategias pedagógicas adaptadas es crucial para mejorar el aprendizaje de conceptos matemáticos. Para hacer frente a esos desafíos, los autores propusieron algunas adaptaciones necesarias en la enseñanza como la incorporación de la lengua de señas como recurso principal de mediación, para asegurarse de que el contenido no se pierda en la traducción o una dependencia exclusiva del texto escrito. También propusieron el uso de las estrategias visuales y gráficas que sirvieran para apoyar la comprensión de conceptos abstractos y hacer un desglose paso a paso de instrucciones para así evitar la sobrecarga de información en un solo momento.

Específicamente, Mousley y Kelly (1998) proponen tres estrategias pedagógicas adaptadas: 1) la comprensión, que consiste en pedir a los estudiantes que expliquen lo entendido en lengua de señas o de forma gráfica, permitiendo al docente identificar vacíos conceptuales; 2)

la visualización, que aprovecha el canal visual para representar gráficamente los problemas mediante esquemas, diagramas o animaciones antes de resolverlos; y 3) el seguimiento de instrucciones, en la que el docente modela el paso a paso de la solución para que los estudiantes lo repliquen con seguridad. Los resultados de este estudio mostraron mejoras significativas en la comprensión. Los estudiantes lograron superar las dificultades con problemas abstractos y replicar procedimientos hasta llegar a soluciones acertadas. Dada la relación entre la lógica de resolución de problemas matemáticos y problemas de programación, las estrategias propuestas en estos estudios pueden aprovecharse para el diseño de un ambiente de aprendizaje de programación de computadores.

Por su parte, Gehret y Elliot (2025), sugieren estrategias de aprendizaje para las personas con discapacidad auditiva mediante el uso estratégico de la redundancia, integrando simultáneamente lengua de señas, subtítulos y apoyos visuales para facilitar la recuperación de información perdida y reforzar la comprensión técnica; y la segmentación, que consiste en dividir las temáticas en fragmentos manejables.

En el campo de la programación, surgen dificultades adicionales para los estudiantes sordos que se combinan con barreras lingüísticas. Tal es el caso del uso de palabras clave en inglés (ejemplo: *if*, *else*, *print*) que exige una sintaxis estricta donde un error mínimo puede causar que el programa no funcione y explicaciones basadas en su mayoría por texto escrito. Esto

provoca que los estudiantes sordos con menos experiencia lectora enfrenten mayores complicaciones para comprender instrucciones y cometan errores al codificar.

Frente a este panorama, el uso de tecnologías digitales abre nuevas posibilidades para la educación inclusiva. Los entornos virtuales de aprendizaje ofrecen una oportunidad para integrar recursos orientados a la población sorda, que normalmente no están disponibles en las aulas presenciales. Entre estos recursos se destacan los videos en lengua de señas, los glosarios visuales y la organización de contenidos en secuencias pequeñas y claras. En este sentido, Hisyamuddin y Tasir (2020) presentaron una plataforma con videos en lengua de señas incorporados directamente en el contenido. A partir de un diseño centrado en el usuario, encontraron que, aunque la usabilidad fue moderada, los estudiantes que accedieron con mayor frecuencia a estos videos lograron mejoras académicas significativas. Este tipo de recursos demostraron que la combinación de lengua de señas, texto simplificado y elementos visuales, junto con contenido segmentado y controles de ritmo, potencia la comprensión y el aprendizaje. Desde esta perspectiva, los entornos virtuales de aprendizaje no solo reducen barreras lingüísticas y comunicativas, sino que también representan un espacio flexible y adaptable que favorece la inclusión educativa.

Este trabajo de grado para optar al título de Licenciada en Diseño Tecnológico se propuso diseñar, implementar y evaluar un ambiente virtual para el aprendizaje de conceptos básicos de

programación en Python, incorporando estrategias apropiadas para las personas con discapacidad auditiva. Específicamente, se seleccionaron las estrategias de comprensión, visualización y seguimiento de instrucciones propuestas por Mousley y Kelly (1998), y citadas por múltiples estudios posteriores. Con la integración de estas estrategias a través del diseño del entorno virtual de aprendizaje, se buscó que los estudiantes sordos comprendieran progresivamente cada concepto computacional antes de avanzar hacia conceptos más complejos. Además, se propuso reforzar el aprendizaje a través de representaciones visuales que faciliten la comprensión e interpretación de conceptos computacionales abstractos como lo son las variables, los tipos de datos y las condicionales; así como organizar secuencias claras que favorezcan el seguimiento de instrucciones paso a paso.

En este sentido, el diseño de este ambiente pretende facilitar la introducción de los estudiantes sordos en los conceptos básicos de programación en Python, promoviendo la comprensión de las variables como contenedores para guardar información y su representación en código; diferenciar entre tipos de datos comunes como números (*integers*) mediante ejemplos visuales y medir la capacidad de toma de decisiones lógicas mediante la estructura *if-else* y la predicción de resultados de salida.

En atención a la problemática expuesta, la pregunta que orientó esta investigación fue:

¿Qué elementos se necesitan para diseñar un ambiente de aprendizaje virtual accesible para enseñar programación en Python a estudiantes sordos de grado décimo del Colegio Público Isabel Segunda de la localidad de Kennedy en Bogotá?

### **1.3 Objetivos**

#### ***1.3.1 Objetivo general***

Diseñar, implementar y evaluar un ambiente de aprendizaje virtual accesible para la enseñanza de programación en Python a estudiantes sordos del curso 1002 de grado décimo del Colegio Público Isabel Segunda de la localidad de Kennedy, en Bogotá.

#### ***1.3.2 Objetivos específicos***

- Identificar las principales barreras físicas, de comunicación, actitudinales y pedagógicas que enfrentan los estudiantes sordos del curso 1002 de décimo grado al aprender programación en Python.
- Analizar cómo la falta de accesibilidad y adaptación de los recursos educativos afecta el proceso de aprendizaje de la programación en Python para los estudiantes sordos del curso 1002.

- Diseñar e implementar estrategias pedagógicas y tecnológicas basadas en recursos visuales y Lengua de Señas Colombiana para la enseñanza de los conceptos básicos de programación en Python en un entorno virtual de aprendizaje.
- Evaluar el uso del ambiente virtual de aprendizaje en la comprensión de conceptos básicos de programación en Python en estudiantes sordos del curso 1002.

## **2. Estado del arte y Marco teórico**

### **2.1 Educación inclusiva y accesibilidad en población sorda**

La educación inclusiva busca garantizar que todos los estudiantes puedan participar en igualdad de condiciones dentro de los procesos de enseñanza y aprendizaje, independientemente de sus características o necesidades educativas. Por esta razón, la UNESCO (2020) señala que todavía existen barreras que limitan el acceso y la permanencia de estudiantes con discapacidad dentro de los sistemas educativos, especialmente cuando las metodologías y recursos no responden a sus formas de comunicación y aprendizaje. En Colombia, el Ministerio de Educación Nacional (2017) y el Instituto Nacional para Sordos (INSOR, 2022) resaltan la importancia de implementar estrategias accesibles que favorezcan la participación de la población sorda mediante apoyos visuales, recursos bilingües y el reconocimiento de la Lengua de Señas Colombiana como parte fundamental de los procesos educativos.

Desde el Diseño Universal para el Aprendizaje (DUA), se plantea la necesidad de ofrecer múltiples formas de presentar la información, participar y expresar el aprendizaje, con el fin de responder a la diversidad presente en el aula (CAST, 2021). Este enfoque reconoce que no todos los estudiantes aprenden de la misma manera, por lo que es importante incorporar recursos visuales, interactivos y multimodales que faciliten la comprensión de los contenidos. En el caso

de los estudiantes sordos, estas estrategias cobran mayor importancia debido a que gran parte del aprendizaje depende de lo visual y de apoyos que reduzcan las barreras del lenguaje escrito.

## **2.2 Pensamiento computacional y enseñanza de programación**

El pensamiento computacional se relaciona con la capacidad de resolver problemas utilizando procesos de lógica, secuenciación, reconocimiento de patrones y construcción de soluciones paso a paso.

Brennan y Resnick (2012) plantean que el pensamiento computacional no debe entenderse únicamente como aprender a programar, sino como un proceso que involucra conceptos computacionales, prácticas computacionales y perspectivas computacionales. Los autores explican que los estudiantes desarrollan pensamiento computacional cuando experimentan, crean proyectos digitales, corrigen errores y construyen soluciones dentro de entornos interactivos. Desde esta perspectiva, la programación permite fortalecer habilidades de lógica, resolución de problemas y creatividad mediante experiencias prácticas y visuales.

De igual manera, Shute, Sun y Asbell-Clarke (2017) destacan que habilidades como la abstracción, la descomposición de problemas y el diseño algorítmico forman parte esencial del pensamiento computacional y pueden fortalecerse con experiencias de programación guiadas y contextualizadas.

En relación con la enseñanza de programación, Severance (2016) plantea que Python es uno de los lenguajes más adecuados para procesos de aprendizaje inicial debido a su sintaxis sencilla y fácil de comprender. Esto permite que los estudiantes puedan enfocarse más en la lógica y la resolución de problemas que en la complejidad del lenguaje. Además, la programación favorece el desarrollo de pensamiento computacional, ya que exige organizar instrucciones, identificar patrones y construir soluciones secuenciales para resolver diferentes situaciones.

### **2.3 Estrategias de enseñanza-aprendizaje efectivas para población sorda**

Las estrategias de resolución de problemas se definen como un conjunto de pasos necesarios para resolver problemas que ocurren al lograr un determinado objetivo (Bransford y Stein, 1984). Estas estrategias de resolución de problemas también se enfatizaron en el paradigma del constructivismo, que se vuelve importante cuando el rol del estudiante cambia de la adquisición de conocimiento a la construcción de conocimiento (Vygotsky, 1978). En pocas palabras, el conocimiento no se ve como los atributos exactos del mundo o un entorno particular, sino más bien como una especie de proceso de construcción por parte de los propios individuos.

Para alentar a los estudiantes sordos a participar en actividades de resolución de problemas, pueden adoptarse las estrategias de resolución de problemas desarrolladas por Mousley y Kelly (1998), quienes identificaron tres estrategias fundamentales que fueron

utilizadas para la enseñanza de matemáticas a estudiantes sordos y que pueden adaptarse al ámbito de la programación.

### ***2.3.1 Comprensión***

Como primera estrategia está la comprensión, la cual consiste en pedir a los estudiantes lo que han entendido de un problema planteado explicándolo ya sea en lengua de señas o de una forma gráfica. La posibilidad de expresarse en su primera lengua el proceso que necesitan y los pasos realizados para dar respuesta al problema refuerzan la comprensión y esto facilita que el docente identifique vacíos conceptuales. Esto es importante para los estudiantes sordos ya que suelen tener menos oportunidades de acceder a explicaciones en su lengua natural, lo que genera que no se pueda comprender lo que se les plantea y menos dar una respuesta a ello.

### ***2.3.2 Visualización***

Es la segunda estrategia en la que se aprovecha la parte visual de las personas sordas, en la que se promueve que antes de resolver un problema lo representen gráficamente o lo organicen en esquemas. Con esta estrategia se busca planificar los pasos de la solución y conectar ideas abstractas con representaciones concretas. En programación puede representarse mediante diagramas de flujo o animaciones que muestren la ejecución del código. La visualización contribuye a cerrar brechas lingüísticas y facilitar la comprensión.

### ***2.3.3 Seguimiento de instrucciones***

La tercera estrategia, busca que el rol docente sea central ya que él es quien modela la resolución de un problema mostrando el paso a paso. Los estudiantes sordos replican el proceso siguiendo a detalle la demostración dada. En programación esto equivale a mostrar en tiempo real la construcción de un programa en código para que luego se pueda modificar con confianza por los estudiantes. Esta estrategia reduce la incertidumbre y brinda seguridad en la aplicación de los conceptos.

Los resultados que arrojó el estudio de Mousley y Kelly (1998) mostró mejoras significativas: la estrategia de comprensión fortaleció el dominio conceptual al poder explicar los procesos en lengua de señas, la visualización permitió superar las dificultades con problemas abstractos y el seguimiento de instrucciones ayudó a que los estudiantes replicaran el código con soluciones acertadas, no se puede olvidar que se hace mención de que es necesario tener una práctica constante y la adaptación del docente para poder enfrentarse a problemas más complejos. A partir de las estrategias planteadas, también se identificaron otras acciones complementarias para favorecer la participación y comprensión de los estudiantes sordos durante procesos de resolución de problemas y programación. Teniendo en cuenta trabajos como los de Gehret y Elliot (2025), que sugieren estrategias de aprendizaje para las personas con discapacidad auditiva mediante el uso estratégico de la redundancia, integrando simultáneamente

lengua de señas, subtítulos y apoyos visuales para facilitar la recuperación de información perdida y reforzar la comprensión técnica; y la segmentación, que consiste en dividir las temáticas en fragmentos manejables.

#### ***2.3.4 Involucrar a un observador par***

Es necesario que participe un observador par, en el caso de que los problemas planteados deban explicarse mediante el lenguaje de señas, y cualquier comprensión resultante de la comunicación en lenguaje de señas se transforme en una solución en forma escrita. Esto permite evaluar si esta estrategia pudiera afectar su explicación en lenguaje de señas y por escrito.

Esta estrategia consiste en incluir un observador par (otro estudiante, que puede o no ser sordo) en el proceso de resolución de problemas. El observador se encarga de explicar los problemas dados utilizando la lengua de señas. Luego, el estudiante sordo debe transformar esa explicación en una solución escrita.

La intención es evaluar cómo el uso del lenguaje de señas influye en la comprensión del problema y en la capacidad de expresar esa comprensión por escrito. Este enfoque ayuda a ver si la comunicación a través de la lengua de señas mejora la capacidad del estudiante para articular una solución de manera escrita.

Esta estrategia no solo mejora la capacidad de traducción de la lengua de señas al español escrito, sino que también promueve la colaboración entre pares, lo que puede enriquecer el proceso de aprendizaje.

### ***2.3.5 Visualización del proceso de resolución de problemas***

Los estudiantes sordos necesitan visualizar lo que se requiere en un proceso de resolución de problemas de principio a fin. Esto es vital para garantizar que los estudiantes puedan desarrollar la estrategia reflexiva necesaria para resolver los problemas que se les presentan. Se utiliza una estrategia de visualización para determinar si la cantidad de formas involucradas en la solución de un problema podría afectar el desempeño de los estudiantes en su conjunto. Esto es importante porque, en la primera estrategia, aunque algunos estudiantes puedan tener buenas habilidades de explicación a través del lenguaje escrito y lengua de señas, esto no significa necesariamente que puedan implementarlas utilizando estrategias reflexivas.

Los estudiantes sordos deben visualizar todo el proceso de resolución de problemas, desde principio a fin. Esto implica que deben imaginar y planificar cada paso necesario para llegar a la solución antes de intentar resolver el problema.

La visualización es crucial porque permite a los estudiantes organizar y estructurar sus pensamientos, lo que es vital para desarrollar una estrategia eficaz de resolución de problemas.

Esta estrategia también investiga si la cantidad de métodos posibles para resolver un problema influye en el rendimiento de los estudiantes.

Aunque un estudiante pueda tener habilidades para explicar un problema usando el lenguaje de señas o la escritura, esta estrategia ayuda a asegurarse de que también puede implementar una solución efectiva mediante un enfoque reflexivo y organizado.

### ***2.3.6 Atención a las instrucciones del profesor durante la codificación***

Los estudiantes sordos deben prestar atención a su maestro respecto a cada uno de los pasos necesarios para resolver el problema que se les presenta. Esta estrategia alienta a los estudiantes sordos a traducir las instrucciones en lengua de señas dadas sus maestros en acciones concretas necesarias para resolver los problemas, en lugar de simplemente escribirlos.

Durante las lecciones de programación en Python, el profesor utilizará el lenguaje de señas para explicar cada paso necesario para resolver un problema de programación. Los estudiantes sordos deben seguir estos pasos traduciendo las instrucciones en código Python, en lugar de solo escribir notas.

Esta estrategia se enfoca en que los estudiantes sordos no solo memoricen las instrucciones del profesor, sino que las apliquen activamente mientras codifican. Esto es vital en Python, donde seguir la lógica paso a paso es esencial para evitar errores de sintaxis y lógica.

Esta estrategia busca desarrollar la capacidad de los estudiantes para aplicar instrucciones en la práctica, lo que es esencial en la programación. Refuerza el aprendizaje activo y la aplicación inmediata de conocimientos teóricos en la escritura de código.

Kaewkamnerd y Suwannarat (2025) destacan que las estrategias visuales e interactivas favorecen el desarrollo del pensamiento computacional en estudiantes sordos, ya que permiten fortalecer procesos de lógica, secuenciación y resolución de problemas mediante recursos accesibles y apoyos gráficos. Esto demuestra la importancia de diseñar ambientes de aprendizaje que respondan a las características visuales y comunicativas de esta población.

#### **2.4 Ambientes virtuales de aprendizaje**

Los ambientes virtuales de aprendizaje se han convertido en una alternativa importante para fortalecer procesos educativos más accesibles e inclusivos. Estos permiten integrar videos, imágenes, actividades interactivas y diferentes formas de representación de la información que favorecen la comprensión de contenidos complejos.

En el caso de estudiantes sordos, Hisyamuddin y Tasir (2020) encontraron que la incorporación de videos en lengua de señas, contenido segmentado y apoyos visuales dentro de las plataformas virtuales mejora la comprensión y el rendimiento académico. Este tipo de

recursos permite reducir barreras lingüísticas y facilita procesos de aprendizaje más autónomos y accesibles.

Desde la teoría sociocultural de Vygotsky (1978), el aprendizaje ocurre mediante procesos de interacción y mediación que permiten al estudiante avanzar desde aquello que puede hacer de manera autónoma hacia niveles de comprensión más complejos con apoyo pedagógico. Este proceso se conoce como Zona de Desarrollo Próximo. En esta investigación, el uso de la Lengua de Señas Colombiana, imágenes, Gifs y ejemplos visuales funcionó como una forma de mediación que facilitó la comprensión de conceptos básicos de programación en Python y permitió acompañar el proceso de aprendizaje de los estudiantes sordos.

### **3. Metodología**

#### **3.1 Enfoque metodológico**

La presente investigación se desarrolla bajo el marco de la Investigación Basada en Diseño (McKenney & Reeves, 2014), enfoque que permitió abordar el problema educativo desde la creación y validación de un objeto tecnológico bilingüe, integrando la teoría pedagógica con la práctica en el aula a través de ciclos de diseño y evaluación. Bajo este modelo, la investigación no se limita a la entrega de un producto, sino que busca comprender cómo el diseño de las herramientas tecnológicas puede potenciar los procesos de aprendizaje en contextos específicos, como es el caso de la población sorda del Colegio Isabel Segunda, integrando teoría pedagógica con lo que realmente ocurre durante su uso en el aula. El proceso incluyó mediciones pretest y posttest aplicadas a un solo grupo, dentro de la Investigación Basada en Diseño.

La Investigación Basada en Diseño es una metodología orientada a la innovación educativa que busca transformar situaciones problemáticas reales mediante la introducción de elementos nuevos como programas, materiales o estrategias— fundamentados en teorías científicas. Se define como un proceso sistemático, flexible, colaborativo e iterativo que busca tanto la resolución de problemas prácticos en contextos auténticos como la generación de conocimiento teórico y principios de diseño. Sus fases fundamentales incluyen el análisis de la

situación y definición del problema, el desarrollo de soluciones de acuerdo con una fundamentación teórica, la implementación en la práctica, la validación o evaluación, y la producción de documentación y principios de diseño, todo ello articulado a través de ciclos continuos de diseño, análisis y rediseño (de Benito y Salinas, 2016).

En el estudio se combinaron datos cuantitativos y cualitativos para analizar el impacto del ambiente virtual de aprendizaje en los estudiantes. Los datos cuantitativos se recolectaron mediante una prueba que fue aplicada antes y después de la intervención. También se analizaron también los registros sobre el número de intentos y los tiempos de ejecución de los contenidos. Por otro lado, se realizó un análisis cualitativo basado en la observación del comportamiento de los estudiantes durante la interacción con el ambiente de aprendizaje, permitiendo identificar dificultades, estrategias de solución y la interacción entre pares.

### **3.2 Descripción de la intervención**

La intervención pedagógica se desarrolló a través de un proceso sistemático que integró la elección de las estrategias de aprendizaje más adecuadas, respaldadas sólidamente por los artículos de investigación revisados y que han sido probadas como efectivas para el diseño e implementación de entornos de aprendizaje mediados por tecnología para estudiantes con discapacidad auditiva, el diseño e implementación del ambiente de aprendizaje, la mediación

lingüística y el trabajo de campo. A continuación, se detallan las etapas que permitieron la ejecución del proyecto:

### ***3.2.1. Elección de estrategias***

El proceso inició con la revisión de literatura especializada en el aprendizaje de personas sordas. Se identificó el estudio de Mousley y Kelly (1998), y su selección de las estrategias de comprensión, visualización y el seguimiento de instrucciones. Estas estrategias son frecuentemente citadas por estudios hasta la actualidad por lo que estas fueron fundamentales para definir la arquitectura pedagógica del contenido, permitiendo que conceptos abstractos de programación se transformaran en recursos visuales comprensibles para la población sorda.

### ***3.2.2. Diseño pedagógico***

Cada una de las tres estrategias de aprendizaje fue diseñada para ser incorporada al ambiente de aprendizaje y para cada una de ellas se diseñaron las actividades y los ejemplos específicos, integrando la Lengua de Señas Colombiana (LSC). El diseño del ambiente virtual de aprendizaje incluyó el diseño del cuestionario de conocimientos básicos en programación que se aplicó al comienzo y al final de la intervención.

### ***3.2.3. Programación del ambiente virtual de aprendizaje***

Con la estructura pedagógica clara, se procedió a la programación del Ambiente Virtual de Aprendizaje (AVA). En este entorno se integraron los retos de código en Python con recursos multimedia en LSC, Gifs, imágenes que permitieron explicar conceptos en los diferentes ejemplos y ejercicios.

#### ***3.2.4. Prueba piloto***

Antes de iniciar el trabajo formal, se llevó a cabo una prueba piloto de reconocimiento del ambiente con otros estudiantes con discapacidad auditiva, lo que permitió verificar la usabilidad de la interfaz y realizar ajustes técnicos antes de la recolección de datos.

#### ***3.2.5. Gestión administrativa y consentimientos***

En cuanto a la gestión institucional, se formalizaron los permisos con la coordinación y los docentes del Colegio Isabel Segunda. Se asignaron para el trabajo los cursos 1102 y 1002, dado que el tema de programación en Python coincidía con su plan de estudios, entonces las intervenciones se llevaron a cabo durante las clases normales de tecnología. Se aplicaron los consentimientos informados de forma oral, con ayuda de la intérprete y el docente a cargo de la clase de informática, explicando a los estudiantes el propósito de la actividad y la importancia de su participación.

### ***3.2.6. Detalle de las sesiones de campo***

El trabajo de campo se consolidó en sesiones presenciales distribuidas de la siguiente manera:

***Sesión 1. Reconocimiento del entorno:*** Los estudiantes exploraron el aplicativo e ingresaron sus datos para familiarizarse con la interfaz, la navegación y la dinámica de respuesta.

***Sesión 2. Pretest y fundamentación técnica:*** Se inició la recolección de datos con el pretest. Luego, los estudiantes navegaron por el bloque de variables. En esta etapa fue clave el apoyo de la intérprete para resolver dudas sobre la sintaxis de Python y la ubicación de signos especiales en el teclado.

***Sesión 3. Aplicación de temas y cierre:*** Se abordaron los contenidos de tipos de datos y estructuras condicionales. Al finalizar, se aplicó el postest para medir el avance del aprendizaje y se realizó un cierre reflexivo con los estudiantes sobre el proceso.

### **3.3 Muestra del estudio**

La presente investigación se llevó a cabo en el Colegio Isabel Segunda (IED), una institución educativa distrital de carácter oficial, situada en la localidad de Kennedy, en Bogotá. La institución se encuentra inmersa en un sector clasificado en estrato socioeconómico 2, atendiendo a una población que, en su mayoría, reside en zonas aledañas con diversas particularidades sociales y económicas. El colegio se destaca por sus procesos de inclusión,

permitiendo que estudiantes con discapacidad auditiva accedan al currículo regular mediante el acompañamiento de intérpretes.

La muestra para este estudio fue seleccionada bajo un criterio de conveniencia, centrada en los grados de educación media. Inicialmente, se contó con la participación de estudiantes de los cursos 1002 y 1102. Es importante mencionar que, aunque en el curso 1102 se encontraban cinco estudiantes sordos matriculados, dos de ellos manifestaron de manera voluntaria su decisión de no finalizar las sesiones de trabajo con el aplicativo. Respetando su autonomía y siguiendo los protocolos éticos, estos estudiantes fueron excluidos del análisis, consolidando una muestra definitiva de seis participantes.

Este grupo de estudio se caracteriza por una distribución equitativa de género, conformada por tres (3) hombres y tres (3) mujeres, con un rango de edad que oscila entre los 16 y los 19 años. Todos los participantes son usuarios de la Lengua de Señas Colombiana (LSC) como su lengua principal.

A continuación, se detalla la distribución de la muestra por curso:

Curso 1102: Tres 3 estudiantes, quienes se encuentran en la etapa final de su ciclo escolar obligatorio y cuentan con conocimientos previos básicos en el área de tecnología e informática.

Curso 1002: Tres 3 estudiantes, quienes iniciaron el proceso de intervención como parte de su formación técnica y académica del periodo vigente.

Los estudiantes seleccionados comparten un perfil de aprendizaje predominantemente visual, una característica común en la población sorda que fue determinante para el diseño de las estrategias pedagógicas del Ambiente Virtual de Aprendizaje (AVA).

**Tabla 1**

*Perfil sociodemográfico de los participantes.*

Estudiante ID	Género	Edad	Curso
20260218112145-5	Femenino	18	1102
20260218104318-4	Femenino	19	1102
20260218104307-2	Masculino	16	1102
20260213105953-0	Femenino	16	1002
20260213105914-1	Masculino	16	1002
20260213105359-7	Masculino	16	1002

*Nota.* Datos obtenidos durante el proceso de investigación. Elaboración propia.

Como se observa en la Tabla 1, la muestra se distribuye con una igualdad en cuanto a género ya que estuvo conformada por un 50% de participantes femeninas y un 50% de participantes masculinos. El rango de edad de los participantes oscila entre los 16 y 19 años.

### **3.4 Instrumentos de recolección de información**

Para dar cumplimiento a los objetivos de la investigación y evaluar la efectividad del ambiente virtual de aprendizaje, se diseñaron instrumentos alineados con el perfil de aprendizaje predominantemente visual de la población sorda. Estos permitieron recolectar datos tanto cuantitativos como cualitativos sobre el proceso de apropiación de conceptos de programación.

#### ***3.4.1 Prueba diagnóstica y de salida (pretest y postest)***

Se diseñó un cuestionario estructurado de 9 ítems con una interfaz gráfica adaptada, donde el estudiante interactúa mediante acciones de "arrastrar y soltar" (drag and drop). Según Mayer (2001), este tipo de interacción favorece el aprendizaje multimedia al reducir la carga cognitiva extraña, permitiendo que el estudiante se concentre en la asociación lógica de los conceptos de Python mediante la manipulación directa de elementos visuales.

El objetivo del pretest es identificar cómo los estudiantes piensan, interpretan y asocian los fundamentos técnicos antes de la intervención. La estructura técnica de la prueba se distribuye en cuatro categorías conceptuales como se observa en la Tabla 2.

**Tabla 2.***Estructura del pretest y el postest.*

Categoría Conceptual	Número de Ítems	Objetivo de la Evaluación
Generalidades	1	Identificar el reconocimiento de Python como un lenguaje de programación y su función básica.
Variables	3	Evaluar la comprensión de las variables como contenedores para guardar información y su representación en código.
Tipos de Datos	2	Diferenciar entre tipos de datos comunes como números ( <i>integers</i> ) mediante ejemplos visuales.
Condicionales	3	Medir la capacidad de toma de decisiones lógicas mediante la estructura if-else y la predicción de resultados de salida.

*Nota.* Elaboración propia basada en el diseño del Ambiente Virtual de Aprendizaje (2026).

### **3.4.2 Diario de campo y observación directa**

Dado el carácter cualitativo y humano de la investigación, la observación directa se constituyó como el instrumento de mayor relevancia para el análisis de los procesos de aprendizaje. Se utilizó el diario de campo para registrar de manera sistemática las interacciones en el aula. Este instrumento permitió documentar aspectos que las pruebas cuantitativas no capturan, tales como:

Asociación bilingüe: Como los participantes vinculaban lo expuesto en LSC con la sintaxis escrita de Python.

Estrategias de resolución: Las rutas lógicas que tomaban los estudiantes ante los retos de código y las dificultades presentadas durante las sesiones presenciales en el Colegio Isabel Segunda.

Autonomía y colaboración: El nivel de independencia alcanzado gracias al material bilingüe y la interacción entre pares para resolver dudas técnicas.

### ***3.4.3 Registro de métricas del sistema***

Como complemento, el AVA fue programado para registrar automáticamente los intentos y el tiempo que cada estudiante dedicó a resolver los retos. Estos datos permiten triangular los resultados de las pruebas (que muestran un avance del promedio general de 5.33 a 5.83) con el desempeño procedimental observado, permitiendo una visión integral del impacto del ambiente virtual en la muestra.

## **3.5 Análisis de la información**

El análisis de la información se realizó integrando tres fuentes principales: los resultados del pretest y posttest, la observación directa y los datos obtenidos del sistema (intentos).

A partir de estas fuentes se realizó una triangulación de la información, con el fin de contrastar los resultados cuantitativos con lo observado durante las sesiones y así tener una comprensión más completa del proceso de aprendizaje.

Se compararon los resultados del antes y el después de la intervención para identificar posibles cambios en la comprensión de los conceptos de programación en Python. Este análisis permitió ver variaciones en el desempeño en el pretest y el posttest, interpretadas como indicadores del proceso de aprendizaje durante la intervención.

A nivel cualitativo, la información obtenida a partir de la observación directa se organizó en tres categorías: comprensión en LSC, estrategias de resolución de problemas y nivel de autonomía durante el uso del ambiente virtual.

#### **4. Diseño del entorno de aprendizaje**

El ambiente virtual de aprendizaje diseñado para esta investigación integra las estrategias pedagógicas y los diferentes contenidos se puede visualizar de manera completa en la siguiente dirección electrónica: <https://educacion.cordialsaludo.com/?i=2>

Las anteriores estrategias se han puesto en marcha en la enseñanza de algunos conceptos de programación del lenguaje Python, enfocado en personas sordas. Los conceptos por trabajar son los de variables, tipos de datos y condicionales en el lenguaje de programación Python, los cuales permiten realizar ejercicios sencillos. Los recursos visuales en Lengua de Señas Colombiana utilizados en el ambiente fueron tomados del Diccionario Básico de la Lengua de Señas Colombiana (INSOR e Instituto Caro y Cuervo, 2006).

##### **4.1. Estructura de contenidos y estrategias de mediación**

A continuación, se detalla el diseño de cada bloque temático integrando la estrategia pedagógica predominante en cada etapa:

###### *Tema 1. Variables*

Definición técnica: Una variable se define como un nombre simbólico que representa un valor almacenado en la memoria del programa. Pedagógicamente, se introduce mediante la

analogía de una "«caja»" donde se guarda información que puede ser modificada en cualquier momento.

*A. Estrategia de comprensión: Variables*

En esta primera parte se aplica la estrategia de comprensión, cuyo objetivo es que las personas sordas puedan demostrar mediante algún tipo de expresión, ya sea lengua de señas o texto sencillo que es una Variable. Esta estrategia busca de alguna manera confirmar que los estudiantes no solo reciban la información, sino que también puedan explicarla con sus propias palabras o señas, lo cual refuerza el aprendizaje.

Ejemplo estrategia de comprensión: Variables

En este primer ejemplo, la variable es **Animal** y tiene el valor de **Perro** , como se observa en la figura 1.

**Figura1.**

*Interfaz de la estrategia de comprensión concepto variables.*

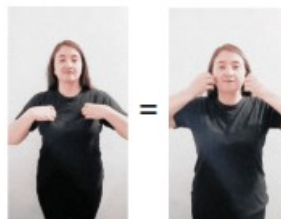
En este primer ejemplo la variable es Animal y tiene el valor de Perro 🐶

Animal = Perro 🐶



Ahora en la variable Animal el valor se cambió por Gato 🐱

Animal = Gato 🐱



*Nota.* Ejemplo elaborado para el ambiente virtual de aprendizaje utilizando recursos visuales y Lengua de Señas Colombiana (LSC). Elaboración propia.

### Ejercicio de la estrategia de comprensión: Variables

En cada ejercicio se pondrá, ya sea la variable o su valor y se debe poner en los espacios en blanco el que corresponda para completar el ejercicio, como en los ejemplos anteriores. En este caso, como se observa en la Figura 2, tenemos la variable deporte= \_\_\_\_\_ , en la parte

inferior se tienen 3 posibles respuestas de las cuales se selecciona una y se arrastra a la casilla vacía.

### Figura 2.

*Ejercicios de la estrategia de comprensión, variables.*

Deporte = \_\_\_\_\_ 

Gato

Animal

Baloncesto

Arrastra aquí tu respuesta 1

*Nota.* Actividad de arrastrar y soltar diseñada para reforzar el concepto de variables mediante recursos visuales. Elaboración propia

#### *B. Estrategia de visualización: Variables*


La segunda estrategia corresponde a la visualización, la cual tiene como propósito que los estudiantes puedan ver de manera clara cómo se crean, se inician y se imprimen las variables en Python. Con esta estrategia se busca apoyar la comprensión del concepto computacional a través de la transferencia lingüística entre la Lengua de Señas Colombiana (LSC), la iconografía y la sintaxis de Python, para que los estudiantes puedan comprender de manera gráfica y concreta la información almacenada en la variable.

### C. Contenido técnico (Python)

Se aborda la sintaxis para la creación de variables, la asignación de valores mediante el operador (=) y el uso de la función *print()* utilizada en la estructura de código y representada por medio de un ojo cómo se ve en la Figura 3, para la salida de datos en pantalla.

#### Figura 3.

*Sentencia Print.*

	<p><b>Print</b></p>	<p>Print es una sentencia.</p> <p>Sentencia: Es una porción de código que representa una orden o acción.</p>
---	---------------------	--

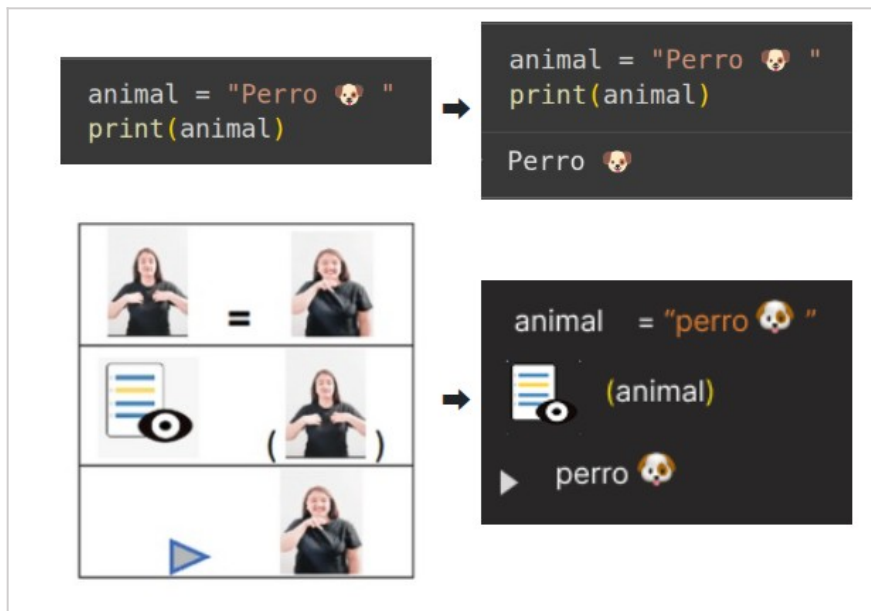
*Nota.* Representación visual de la sentencia print en Python utilizada dentro del ambiente virtual de aprendizaje. Elaboración propia.

#### Ejemplo estrategia visualización: Variables

En este ejercicio se ve la explicación usando código en Python y apoyado de emojis y LSC, (figura 4).

#### Figura 4.

*Interfaz de la estrategia de visualización concepto variables*



*Nota:* La figura muestra como se hace la explicación por medio de imágenes y código en Python sobre el concepto de variable. Fuente propia.

### **Ejemplo explicado por medio de código: Variables**

En este ejemplo se puede ver (figura 5), que ya todo es en código Python en el editor de código, de cómo se ve la secuencia al ejecutarse y la sentencia `print` cumple con su objetivo que es el de mostrar la información.

#### **Figura 5.**

*Código Python de cómo se ven las variables y la LSC al final.*



*Nota.* Ejemplo visual de la función print() en Python dentro del ambiente virtual de aprendizaje. Elaboración propia.

### **Ejercicio de la estrategia de visualización: Variables**

Con estos ejercicios se espera que los estudiantes demuestren que han entendido por medio de la segunda estrategia de visualización, el concepto de variable.

En cada ejercicio se presentará una variable o su valor (Figura 6), y se deberá arrastrar la opción correspondiente a la casilla para completar la estructura, tal como se mostró en los ejemplos anteriores.

### **Figura 6.**

*Ejercicios de la estrategia de visualización, variables.*

Deporte = Boxeo \_\_\_\_\_

Arrastra aquí tu respuesta 5

*Nota.* Actividad interactiva de visualización desarrollada en el ambiente virtual de aprendizaje. Elaboración propia.

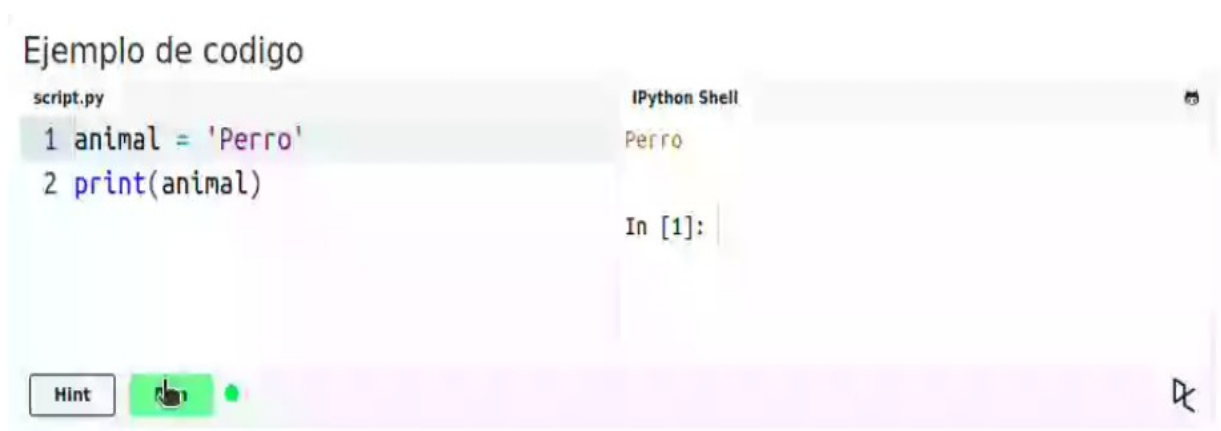
#### *D. Estrategia de seguimiento de instrucciones: Variables*

La tercera estrategia consiste en demostrar paso a paso la escritura de código en un editor en línea de Python, para que los estudiantes puedan replicar las instrucciones en su propio editor traduciendo cada paso en código funcional, promoviendo así el aprendizaje activo y la aplicación

práctica de la lógica de programación. Este paso es esencial para poder comprender y apropiarse la sintaxis del lenguaje y desarrollar la capacidad de seguir procesos secuenciales.

### Figura 7.

*Interfaz estrategia seguimiento de instrucciones para el concepto de variables.*



```
Ejemplo de código
script.py
1 animal = 'Perro'
2 print(animal)

IPython Shell
Perro
In [1]:
```

*Nota.* Representación de la declaración de una variable y visualización de su salida mediante la función `print()` en Python. Elaboración propia..

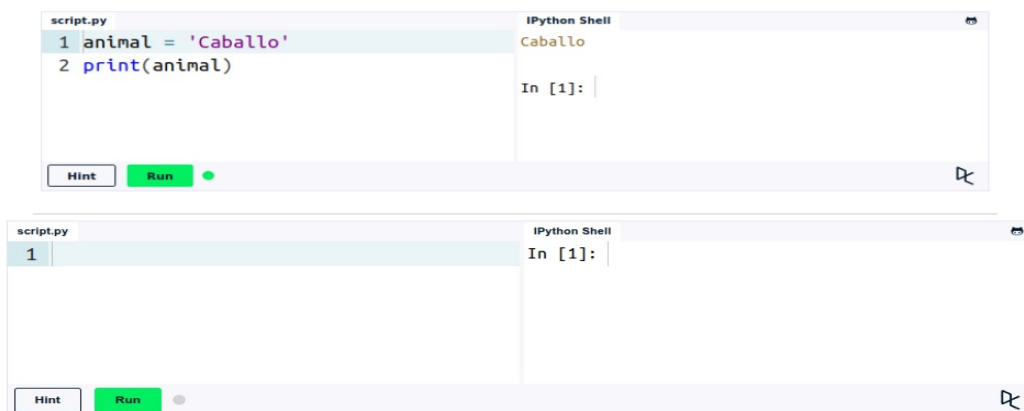
### Ejemplo estrategia seguimiento de instrucciones: Variables

En este ejemplo (Figura 7), el estudiante puede observar que al darle en el botón de run como la estructura de código se ejecuta y muestra en pantalla el valor que se le dio a la variable.

### Figura 8.

*Ejercicio seguimiento de instrucciones para el concepto de variables.*

## Ejercicio 3



```
script.py | IPython Shell
1 animal = 'Caballo'
2 print(animal)
Caballo
In [1]: |

script.py | IPython Shell
1
In [1]: |
```

Nota. Representación de la declaración de otra variable y visualización de su salida mediante la función `print()` en Python. Elaboración propia.

En este ejercicio los estudiantes tienen el ejemplo del primer ejercicio y deben pasar el mismo código escrito al editor y ejecutarlo, Figura 8.

### *Tema 2. Tipos de datos*

Definición técnica: Un tipo de dato es una categoría que define la clase de valor que puede almacenar una variable y las operaciones que se pueden realizar con esos valores.

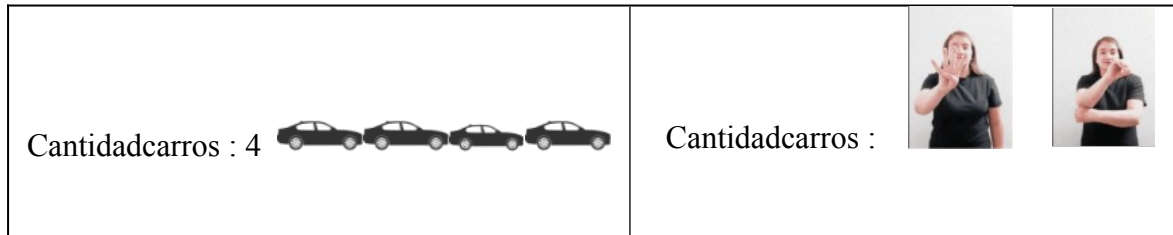
- *Números Enteros (int):*

Los enteros se usan para contar objetos. Son números sin decimales, Figura 9.

Ejemplo: 2, 3,10,25,56,78.

**Figura 9.**

*Explicación tipos de datos con numero entero.*



*Nota:* Ejemplo de tipos de datos con números enteros. Fuente propia.

- *Números decimales (float):*

Los números decimales o de punto flotante son números con decimales, como el precio de un producto o una medición de peso Figura 10.

Ejemplo: 1.66 – 2.26 - 6.87 – 3.14

**Figura 10.**

*Explicación tipos de datos con números decimales.*



*Nota:* Ejemplo de tipos de datos con números decimales. Fuente propia.

- *Booleanos (bool):*

Es un tipo de dato que permite almacenar dos valores True (Verdadero) o False (Falso),

Figura 11.

**Figura 11.**

*Explicación tipos de datos con valores Booleanos.*



*Nota:* Ejemplo de tipos de datos con valores booleanos. Fuente propia.

*A. Estrategia de comprensión: Tipos de datos.*

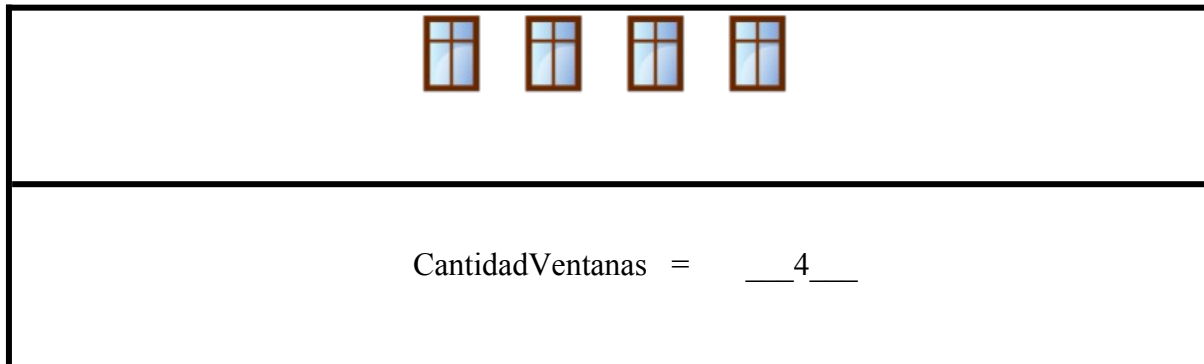
Haciendo uso de esta estrategia se realiza el siguiente ejemplo, donde se hace uso de tipos de datos en forma gráfica, en texto, numero para que se siga trabajando de la misma manera que los ejemplos anteriores y se arrastre la respuesta correcta la cual sería el número cuatro ya que en la imagen se muestran cuatro ventanas a lo que se refiere a números enteros.

### Ejemplo de la estrategia de comprensión: Tipos de datos

#### Figura 12.

*Interfaz de la estrategia de comprensión del concepto de tipos de datos.*

Como se observa en la Figura 12, la explicación de tipos de datos usando imágenes y texto, mostrando además la respuesta correcta.



1,4	2,0	4
-----	-----	---


*Nota:* Fuente propia.

En este ejercicio se puede ver en la Figura 13, el uso de imágenes para hacer la explicación ayuda una perspectiva de unir conocimientos previos y en las respuestas de recordar la explicación anteriormente dada.

## Ejercicio de la estrategia de comprensión: Tipos de datos

**Figura 13.**

*Interfaz estrategia de comprensión ejercicio de tipos de datos.*

		
LlevarBicicleta = _____		
Int	Float	False

*Nota:* Fuente propia

### *B. Estrategia de visualización: Tipos de datos*

La segunda estrategia corresponde a la visualización, la cual tiene como propósito que los estudiantes puedan ver de manera clara cómo se usan los tipos de datos en Python. Con esta estrategia se busca apoyar la comprensión del concepto computacional a través de la transferencia lingüística entre la Lengua de Señas Colombiana (LSC), la iconografía y la sintaxis


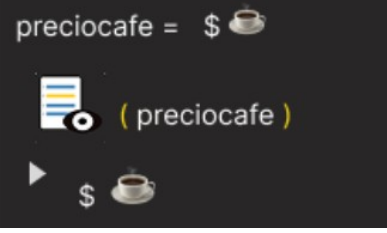

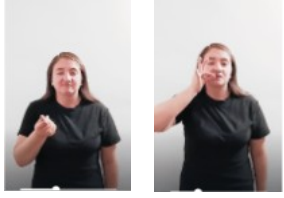
de Python, para que los estudiantes puedan comprender de manera gráfica y concreta la información almacenada en la variable.

### Ejemplo de la estrategia de visualización: Tipos de datos

Se inicia mostrando la estructura del código usando tipos de datos y luego de eso, al ejecutarlo qué valor nos saldrá en pantalla (Figura 14), a su vez se ponen más imágenes que permitan la interpretación de los conceptos.

#### Figura 14.

*Explicación de tipos de datos mediante código Python y LSC.*

<pre>PrecioCafé = 1,50 print (PrecioCafé)</pre>		<pre>PrecioCafé = 1,50 print (PrecioCafé) (1, 50)</pre>
<pre>preciocafe = \$ ☕</pre> 		 <span style="margin-left: 20px;">= 1,75</span>

Nota: Fuente propia.

### Ejercicio de la estrategia de visualización: Tipos de datos

Se usa una imagen que sirve para conectar una estructura y dar respuesta a la pregunta que fue escrita en el editor de código y en la que se puede visualizar cómo se

escribe una variable con un tipo de dato decimal (Figura 15), de lo cual se espera su respuesta de acuerdo con lo que se pregunta.

**Figura 15.**

*Ejercicio tipos de datos usando decimales.*

```
Estatura = 1,66  
print (Estatura)
```

---

2	Café	1.66
---	------	------

*Nota:* Fuente propia

*C. Estrategia de seguimiento de instrucciones: Tipos de datos*

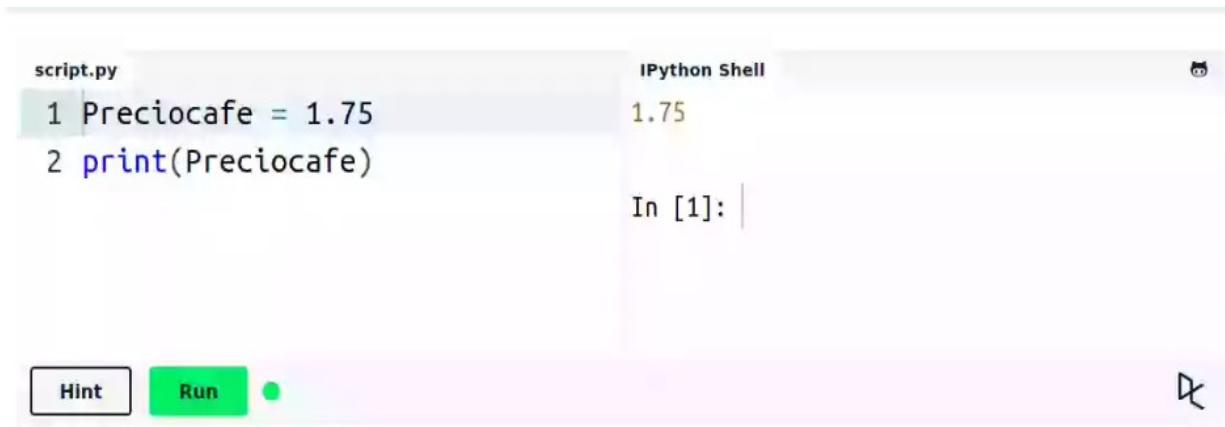
La tercera estrategia consiste en demostrar paso a paso la escritura de código en un editor en línea de Python, para que los estudiantes puedan replicar las instrucciones en su propio editor traduciendo cada paso en código funcional, promoviendo así el aprendizaje activo y la aplicación práctica de la lógica de programación. Este paso es esencial para poder apropiarse de la sintaxis del lenguaje y desarrollar la capacidad de seguir procesos secuenciales.

### Ejemplo de la estrategia de seguimiento de instrucciones: Tipos de datos

Estructuras de código listas para ser ejecutadas y ver su resultado, (Figura 16).

#### Figura 16.

*Interfaz de la estrategia de seguimiento de instrucciones, ejemplo Tipos de datos*



The screenshot shows a code editor interface. On the left, a file named 'script.py' contains two lines of Python code: '1 Preciokafe = 1.75' and '2 print(Preciokafe)'. On the right, the 'IPython Shell' displays the output '1.75' and a prompt 'In [1]:'. At the bottom, there are buttons for 'Hint' and 'Run', and a cursor icon on the right.

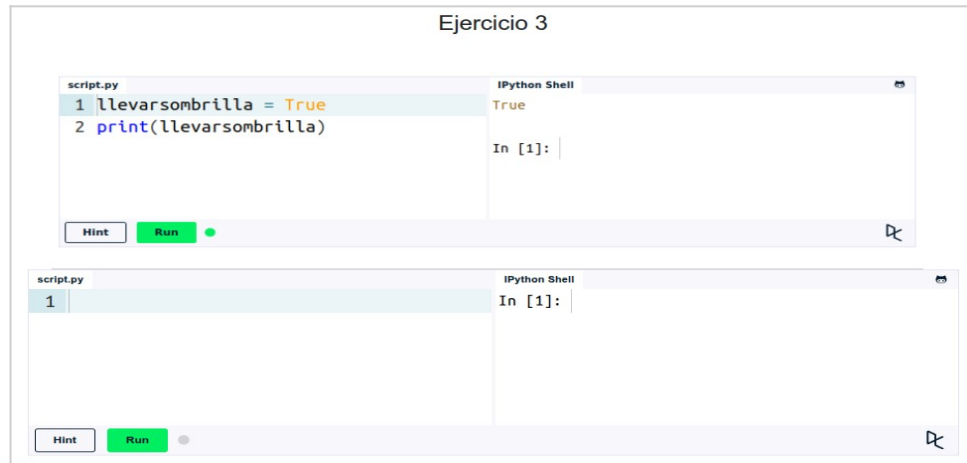
*Nota:* Fuente propia.

### Ejercicio estrategia seguimiento de instrucciones: Tipos de datos

Escribir la estructura de código y ejecutarla en el editor de código, (Figura 17).

#### Figura 17.

*Ejercicio de seguimiento de instrucciones con tipos de datos.*



*Nota:* Fuente propia.

### *Tema 3. Condicionales*

Las estructuras condicionales en Python permiten que un programa tome decisiones basadas en condiciones específicas. Utilizando la declaración `if`, para condición verdadera. Si es falsa, se puede usar `else`, (Figura 18).

#### **Figura 18.**

*Diagramas con apoyo de imágenes sobre la estructura de condicionales.*

```

lloviendo = True

if lloviendo:
    print("Lleva sombrilla")
else:
    print("No necesitas sombrilla")

Lleva sombrilla

```



*Nota:* Fuente propia.

#### *A. Estrategia de comprensión: Condicionales*

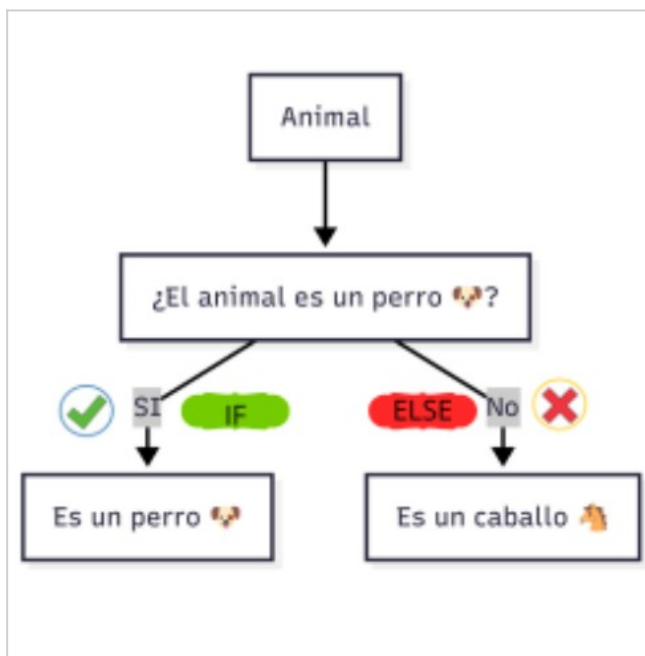
En esta parte se presentan ejemplos aplicando la primera estrategia de comprensión, donde se introducen las condicionales en Python de manera visual. A través de diferentes formas de representación, como texto, imágenes y lengua de señas, se busca que los estudiantes comprendan cómo funcionan las decisiones en un programa.

#### **Ejemplo de la estrategia de comprensión: Condicionales**

Se puede observar un ejemplo que muestra una condicional relacionada con animales, en la que se evidencian los dos posibles caminos: el sí (If), que afirma que el animal es un perro, y el no (Else), que indica que el animal es un caballo (Figura 19). De esta manera, los estudiantes pueden visualizar cómo el programa elige una u otra opción según la condición establecida.

**Figura 19.**

*Estrategia de comprensión, estructura condicional para observar la toma de decisiones.*



*Nota.* Representación gráfica del funcionamiento de las estructuras condicionales if y else en Python. Elaboración propia.

Ahora observamos el mismo ejemplo en la Figura 20, pero acompañado de representaciones en lengua de señas y texto. Este ejemplo busca mostrar lo mismo que el anterior, apoyándose en representaciones en lengua de señas de los animales que aparecen en la condicional, para facilitar la comprensión del concepto.

**Figura 20.**

*Estructura condicional con LSC, imágenes y texto para la toma de decisiones.*



*Nota.* Representación gráfica del funcionamiento de las estructuras condicionales if y else en Python. Elaboración propia.

### *B. Estrategia de visualización: Condicionales*

La segunda estrategia corresponde a la visualización, la cual tiene como propósito que los estudiantes puedan ver de manera clara cómo se inician, implementan, estructuran e imprimen los resultados de las condicionales en Python. Con esta estrategia se busca apoyar la comprensión del concepto computacional a través de la transferencia lingüística entre la Lengua de Señas Colombiana (LSC), la iconografía y la sintaxis de Python, para que los estudiantes puedan comprender de manera gráfica y concreta la información almacenada en la variable.

### **Ejemplo de la estrategia de visualización: Condicionales**

Esta misma condicional se representa en el editor de código, para observar cómo se escribe dentro de esta interfaz (Figura 21). De esta forma, los estudiantes podrán relacionar la representación visual del diagrama con la estructura real del código en Python, comprendiendo cómo funciona el if y el else en un entorno de programación.

### Figura 21.

*Interfaz estrategia de visualización, conceptos condicionales.*



```
animal = "Perro"
if animal == "Perro":
    print("🐶 Es un perro")
else:
    print("🐎 Es un caballo")
```

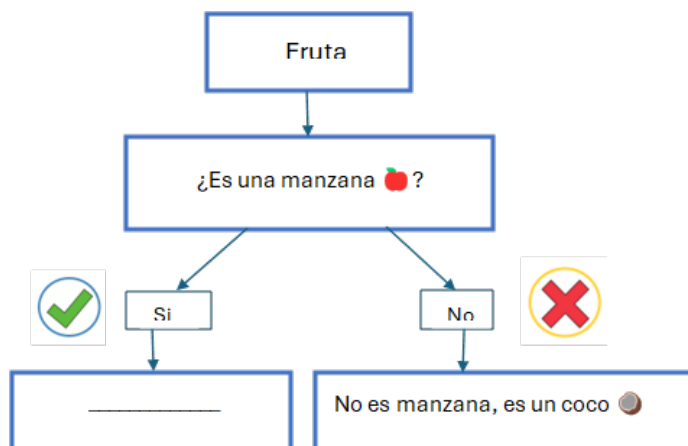
```
animal = "Perro"
if animal == "Perro":
    print("🐶 Es un perro")
else:
    print("🐎 Es un caballo")
🐶 Es un perro
```

### Ejercicio de la estrategia de visualización: Condicionales

Se presentan a continuación varios ejercicios que utilizan las condicionales de forma gráfica, combinando texto y lengua de señas (Figura 22). De esta manera, se mantiene la misma dinámica de los ejercicios anteriores. El estudiante deberá arrastrar la respuesta correcta hacia la casilla correspondiente, teniendo en cuenta la instrucción que se plantea en cada ejercicio.

### Figura 22.

*Estrategia de visualización, ejercicio toma de decisiones con condicionales.*



Es un banano	Es un gato	Es una manzana
--------------	------------	----------------

*Nota:* Ejercicio de estructura condicional. Fuente propia

### *C. Estrategia de seguimiento de instrucciones: Condicionales*

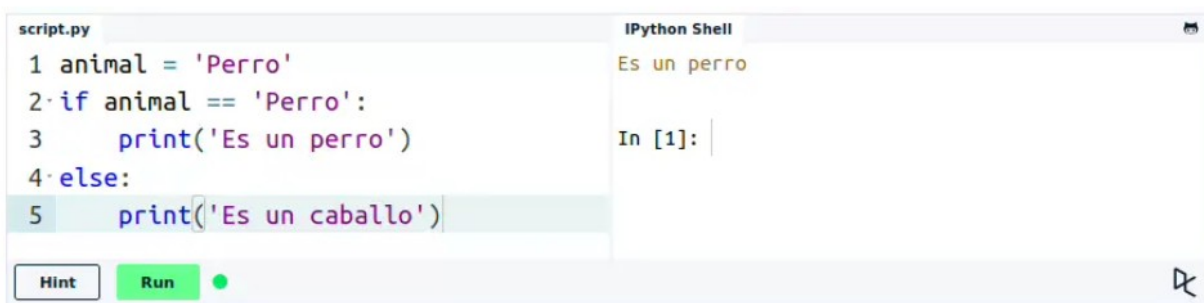
La tercera estrategia consiste en demostrar paso a paso la escritura de condicionales en código en un editor en línea de Python, para que los estudiantes puedan replicar las instrucciones en su propio editor traduciendo cada paso en código funcional, promoviendo así el aprendizaje activo y la aplicación práctica de la lógica de programación. Este paso es esencial para poder apropiarse la sintaxis del lenguaje y desarrollar la capacidad de seguir procesos secuenciales.

### Ejemplo de la estrategia de seguimiento de instrucciones: Condicionales

Estructuras de código escritas para obtener un resultado solo con hacer clic al botón de run (Figura 23).

#### Figura 23.

*Ejemplo de la estrategia de seguimiento de instrucciones*



The image shows a code editor window with two panes. The left pane, titled 'script.py', contains the following Python code:

```
1 animal = 'Perro'
2 if animal == 'Perro':
3     print('Es un perro')
4 else:
5     print('Es un caballo')
```

The right pane, titled 'IPython Shell', shows the output of the code execution: 'Es un perro'. Below the output, it says 'In [1]:' followed by a cursor. At the bottom of the editor, there are two buttons: 'Hint' and 'Run' (with a green dot next to it).

*Nota:* Ejemplo de estructura condicional en el editor de código. Fuente propia.

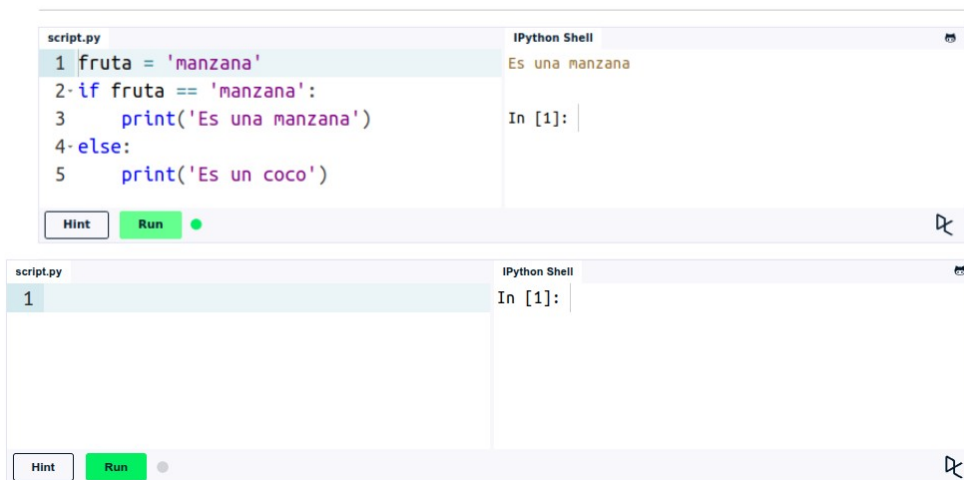
### Ejercicio de la estrategia de seguimiento de instrucciones: Condicionales

En este ejercicio (Figura 24), se muestra la guía de cómo debe ir la estructura y de qué al ejecutarlo nos muestra un resultado y lo que se espera es que el estudiante, de acuerdo con el ejemplo, lo siga, lo escriba y llegue al mismo resultado propuesto.

#### Figura 24.

*Ejercicio de condicionales con la estrategia seguimiento de instrucciones.*

## Ejercicio 3



```
script.py
1 fruta = 'manzana'
2 if fruta == 'manzana':
3     print('Es una manzana')
4 else:
5     print('Es un coco')

IPython Shell
Es una manzana

In [1]: |

Hint Run ●
```

```
script.py
1

IPython Shell
In [1]: |

Hint Run ●
```

*Nota:* Fuente propia.

## 4.2. Arquitectura y desarrollo del sistema

Una vez definido el diseño pedagógico y los recursos del ambiente virtual de aprendizaje, se procedió a su implementación técnica, con el fin de materializar las estrategias propuestas en una solución funcional e interactiva. El sistema fue desarrollado bajo una arquitectura monolítica, en la cual los diferentes componentes (gestión de contenidos, interacción del usuario y almacenamiento de datos) se integran en una única estructura; esta decisión permitió simplificar el desarrollo y facilitar la implementación del entorno en un contexto educativo.

La estructura del ambiente se organiza de manera modular en coherencia con el diseño pedagógico planteado. Cada módulo contiene actividades y recursos asociados a un tema específico de programación, los cuales son almacenados en archivos JSON estáticos. Estos archivos son procesados mediante PHP, permitiendo la carga dinámica de preguntas, ejercicios y contenidos dentro de la interfaz.

En cuanto a la interacción, el sistema cuenta con funciones específicas para la presentación y validación de las actividades. Por un lado, se implementa una función que permite leer y mostrar las preguntas al usuario y, por otro, una función encargada de validar las respuestas mediante el uso de identificadores únicos (ID). Esta validación se articula con las actividades interactivas, como el arrastre de elementos, proporcionando retroalimentación inmediata a través de cambios visuales (*verde para aciertos y rojo para errores*).

Como componente central para la práctica técnica, se integró la herramienta DataCamp Light, una biblioteca basada en JavaScript que permite la ejecución de código Python directamente en el navegador. Esta integración es fundamental, ya que convierte el ambiente en un entorno de programación real, permitiendo que los estudiantes escriban, prueben y ejecuten scripts de Python en una consola interactiva sin salir de la plataforma (DataCamp, s. f.), garantizando así la aplicación inmediata de los conceptos de sintaxis trabajados.

Respecto al almacenamiento de la información, se emplea una base de datos MySQL en la cual se registran variables como el identificador de la pregunta, la respuesta seleccionada, su validez (correcta o incorrecta), el usuario y el momento de la interacción. Esto permite realizar un seguimiento detallado del desempeño de los estudiantes. En términos tecnológicos, el sistema integra PHP para la lógica del servidor, MySQL para la gestión de datos, y tecnologías web estándar como HTML5, CSS3 y JavaScript para la construcción de la interfaz. Por último, para garantizar la identificación de cada usuario, se implementa el manejo de sesiones en el navegador, permitiendo asignar una sesión única a cada estudiante y facilitando el seguimiento individual del proceso de aprendizaje.

## **5. Resultados**

En este capítulo se presentan los resultados obtenidos tras la implementación del ambiente virtual de aprendizaje diseñado para la enseñanza de conceptos básicos de programación en Python, mediado por Lengua de Señas Colombiana (LSC). El objetivo principal de este análisis es evaluar la incidencia de las estrategias pedagógicas de Mousley y Kelly (1998), como son la comprensión, visualización y seguimiento de instrucciones en el proceso de aprendizaje de la población sorda participante.

### **5.1 Información sociodemográfica de la población**

Es necesario describir el perfil de los estudiantes que formaron parte de la muestra. Se trabajó con un grupo de 6 estudiantes pertenecientes a los grados décimo y undécimo del colegio Isabel Segunda. La diferencia en las edades y géneros permite observar la adaptabilidad del ambiente virtual de aprendizaje en diferentes perfiles de usuario dentro del contexto escolar.

**Tabla 3.***Relación de participantes para el análisis de desempeño*

Estudiante ID	Género	Edad	Curso
5	Femenino	18	1102
4	Femenino	19	1102
2	Masculino	16	1102
0	Femenino	16	1002
1	Masculino	16	1002
7	Masculino	16	1002

*Nota.* Datos obtenidos durante el proceso de investigación. Elaboración propia.

## **5.2. Resultados generales en los conceptos de programación**

Una vez definido el perfil de los participantes, se presentan los resultados obtenidos en el proceso de formación. Para evaluar el impacto del ambiente virtual, se aplicó una prueba de conocimientos inicial (pretest) y una prueba final (postest). El instrumento aplicado constó de 9 preguntas diseñadas para identificar la comprensión visual y técnica sobre tres ejes temáticos: variables, tipos de datos y condicionales. La Figura 25 muestra, como ejemplo la primera pregunta de la prueba.

**Figura 25.**

*Primera pregunta que se encuentra en el pretest y el postest.*



*Nota.* Primera pregunta de la prueba inicial pretest y final postest. Fuente propia.

La Tabla 4 presenta los resultados obtenidos en las pruebas pretest y postest en cada uno de los conceptos computacionales evaluados.

**Tabla 4.**

*Resultados de la prueba diagnóstica (Pretest) y prueba final (Postest).*

Estudiante ID	General (1) Pre/Post	Variables (3) Pre/Post	T. Datos (2) Pre/Post	Condicionales (3) Pre/Post	Total Pre/Post
20260218112145-5	1/1	3/3	1/2	1/2	6/8

20260218104318-4	0/0	0/2	1/0	1/0	2/2
20260218104307-2	1/1	3/3	1/2	2/3	7/9
20260213105953-0	1/1	2/3	1/2	1/1	5/7
20260213105914-1	1/0	3/1	2/1	1/1	7/3
20260213105359-7	1/1	1/2	2/2	1/1	5/6
Promedio general	0,83/0,66	2,00/2,33	1,33/1,50	1,16/1,33	5,33/5,83

*Nota:* Resultados obtenidos en el pretest y postest. Fuente propia.

Respecto al desempeño grupal, se evidenció un avance progresivo tras la intervención pedagógica. El promedio obtenido en el pretest fue de 5.33/9, mientras que en el postest la cifra ascendió a 5.83/9. Aunque el incremento numérico es moderado, este ascenso es cualitativamente significativo, ya que demuestra que el ambiente virtual contribuyó positivamente al proceso de aprendizaje de conceptos básicos de programación en Python, permitiendo a los estudiantes superar su base de conocimientos previos.

La mejora entre el pretest y el postest se evidenció en los tres conceptos evaluados. El mayor crecimiento se registró en el concepto de Variables, donde el promedio grupal subió de 2.00 a 2.33. Es relevante destacar que, aunque el concepto de variables se abordó desde una estructura sencilla, la estrategia de mediación se centró en usar la Lengua de Señas (LSC) y

emojis/imágenes para que el estudiante no tuviera que leer una explicación larga en español, sino que viera la relación de inmediato.

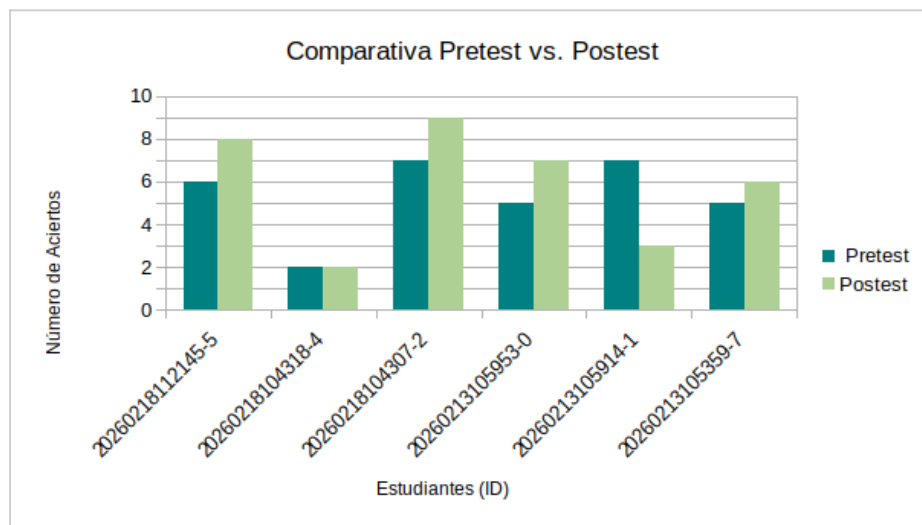
Esta simplificación en la presentación del contenido facilitó la transición de un concepto abstracto a una representación mental clara, sugiriendo que la Lengua de Señas Colombiana (LSC) potencia la capacidad de retención de la sintaxis.

En los tipos de datos y condicionales la ganancia entre el pretest y el posttest fue de 0,17. Al ser estos conceptos más abstractos y complejos, es posible que su comprensión requiera un entrenamiento más prolongado.

La Figura 26 muestra las variaciones en las calificaciones entre el pretest y el posttest en los seis estudiantes que participaron en el estudio.

**Figura 26**

*Grafica que muestra una comparativa entre la prueba inicial y la final.*



*Nota:* fuente propia.

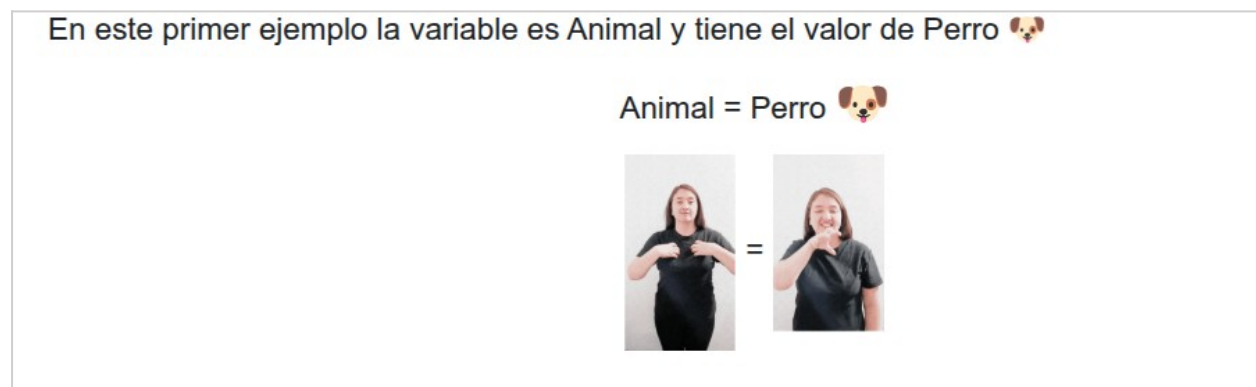
En cuanto al rendimiento individual, el 66% de los participantes (4 de 6 estudiantes) incrementó su puntaje tras la intervención. Este resultado es relevante considerando que los estudiantes se enfrentaron a conceptos computacionales nuevos, por lo que las estrategias implementadas en el entorno virtual de aprendizaje lograron ser efectivas en la mayoría de los casos. Sin embargo, persisten dificultades que no pudieron ser identificadas con profundidad debido al tiempo limitado de la intervención y a que la investigadora no es la docente habitual de las aulas estudiadas.

### 5.3. Resultados de la estrategia de comprensión

La estrategia de comprensión se implementó mediante una mediación que articuló la sintaxis del código en Python con su equivalente en Lengua de Señas Colombiana (LSC), apoyada con imágenes, secuencias visuales y Gifs. Esta estructura permitió establecer una relación directa entre el contenido textual y su interpretación visual, favoreciendo el acceso inicial a los conceptos (Figura 27).

#### Figura 27.

*Ejemplo de mediación (Sintaxis Python - Emoji - Señal LSC).*



*Nota:* Fuente propia

La mediación en Lengua de Señas Colombiana (LSC) no solo permitió traducir el contenido, sino que facilitó la construcción de significado, actuando como un puente entre el lenguaje técnico de la programación y la comprensión de los estudiantes.

Durante las sesiones en el aula, se observó que los estudiantes realizaban un proceso previo de interpretación antes de responder a las actividades. Cuando los contenidos incluían apoyo en LSC, los estudiantes mostraban comprensión a través de expresiones faciales de reconocimiento. En contraste, cuando el contenido implicaba mayor contenido textual, especialmente en estructuras condicionales, se observaron gestos de duda y confusión, lo que sugiere que la dificultad no estuvo únicamente en el concepto, sino en la comprensión del enunciado.

En estos casos, aunque los estudiantes intentaban apoyarse en las imágenes y recursos disponibles, cuando no lograban comprender completamente la actividad, recurrían al ensayo y error como una estrategia alternativa para avanzar.

Este comportamiento coincide con lo observado en el pretest, donde los estudiantes ya presentaban dificultades en la interpretación de los enunciados. Durante la implementación, estas dificultades se observaron en el aula a través de expresiones de duda y confusión, y aunque en el posttest se observaron avances, estos no fueron homogéneos, lo que indica que la comprensión sigue siendo un factor determinante en el aprendizaje.

En relación con el número de intentos, se encontró que cuando la explicación era clara, los estudiantes respondían correctamente en el primer intento. Específicamente, en el concepto

de Variables (Var\_E1), el grupo registró un promedio de apenas 1.08 intentos, lo que indica que el ensayo y error no fue la estrategia inicial, sino una alternativa cuando la comprensión no se lograba completamente.

En cuanto a los contenidos, las variables fueron comprendidas con mayor facilidad, mientras que en los tipos de datos se presentaron dificultades asociadas al uso de términos en inglés. Por su parte, las estructuras condicionales representaron el mayor nivel de dificultad, demostrando que la comprensión del enunciado no siempre se tradujo en la aplicación de la lógica de programación.

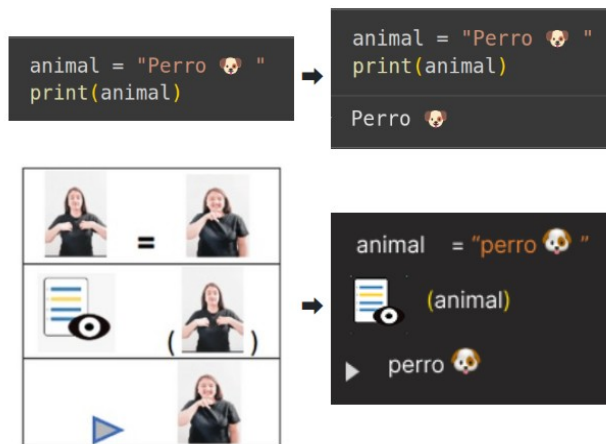
En este sentido, el comportamiento observado refleja que el aprendizaje en esta estrategia estuvo centrado en la comprensión del contenido antes de la acción, lo que la diferencia de las demás estrategias implementadas.

#### **5.4 Resultados de la estrategia de visualización**

La estrategia de visualización se estructuró mediante el uso de recursos gráficos como imágenes, Gif, diagramas y secuencias visuales, organizados en una lógica donde se presentaba el código en Python, su representación visual y su resultado. Esta organización permitió reducir la dependencia del texto y facilitar una interpretación más directa de los contenidos (Figura 28).

### Figura 28.

*Ejemplo de código (Sintaxis Python - Emoji - Seña LSC).*



*Nota:* Fuente propia.

Durante las sesiones, se observó que los estudiantes interactuaban principalmente con los elementos visuales, priorizando la observación de imágenes sobre la lectura del enunciado. En muchos casos, respondían de manera inmediata tras interpretar los recursos visuales, lo que dejó ver que el comportamiento estuvo orientado hacia la respuesta rápida más que hacia el análisis detallado.

#### 5.4.1 Análisis de número de intentos en las estrategias de comprensión y visualización

La Tabla 5 presenta el número de intentos en la resolución de los ejercicios propuestos en las tres unidades desarrolladas el entorno virtual de aprendizaje, durante la aplicación de las estrategias de comprensión (estrategia 1) y visualización (estrategia 2).

**Tabla 5.**

*Número de intentos por estrategia y unidad temática*

ID Usuario	Variables Estrategia 1 (Intentos)	Variables Estrategia 2 (Intentos)	Tipos de datos Estrategia 1 (Intentos)	Tipos de datos Estrategia 2 (Intentos)	Condicionales Estrategia 1 (Intentos)	Condicionales Estrategia 2 (Intentos)
4	4	4	7	3	2	1
2	4	4	14	3	3	2
5	4	3	9	3	2	1
1	4	7	8	3	2	3
7	5	4	13	3	2	2
0	5	4	15	3	2	3
Promedio por intento	4.3	4.3	11	3	2.2	2

*Nota:* Fuente propia

En la unidad de Tipos de datos se observó algo importante, la estrategia de comprensión fue en la que los estudiantes realizaron un número de intentos muy elevado, con un promedio de 11. Esto indica que tratar de entender el concepto sin un apoyo visual fuerte fue el reto más difícil para ellos. Sin embargo, al aplicar la estrategia de visualización, los intentos bajaron drásticamente a un promedio de 3, esto es importante ya que se demuestra que la visualización fue la herramienta que permitió a los estudiantes aprender este concepto de forma ágil.

Con el tema de variables tanto la estrategia de comprensión como la de visualización mostraron un comportamiento estable con un promedio de 4,3 intentos. Esto indica que, para este concepto, las dos formas de enseñanza permitieron una comprensión equilibrada sirviendo como base para los siguientes temas.

En condicionales se registró el número más bajo de intentos con un promedio de 2,2 y 2,0, que, aunque podría parecer un éxito rotundo, las observaciones en el aula indican que esta rapidez surgió del aprendizaje colaborativo y del ensayo y error facilitado por la interfaz, ya que si lo que elegían no era correcto podían corregir inmediatamente. Esto indica que cuando los conceptos son complejos requieren de una mediación más profunda para asegurar que sí hay aprendizaje. Para terminar los resultados confirman que las estrategias de visualización en el ambiente de aprendizaje ayudan a los estudiantes a superar el obstáculo del texto escrito

permitiéndoles resolver ejercicios con más claridad, pero en algunos conceptos de mayor complejidad la imagen por sí sola no es suficiente, ya que el aprendizaje significativo en el estudiante sordo ocurre cuando existe una integración visual que les permite rapidez, pero la explicación en Lengua de Señas Colombiana (LSC) es la que garantiza que el estudiante entendió lo que está haciendo.

### **5.5 Resultados de la estrategia de seguimiento de instrucciones**

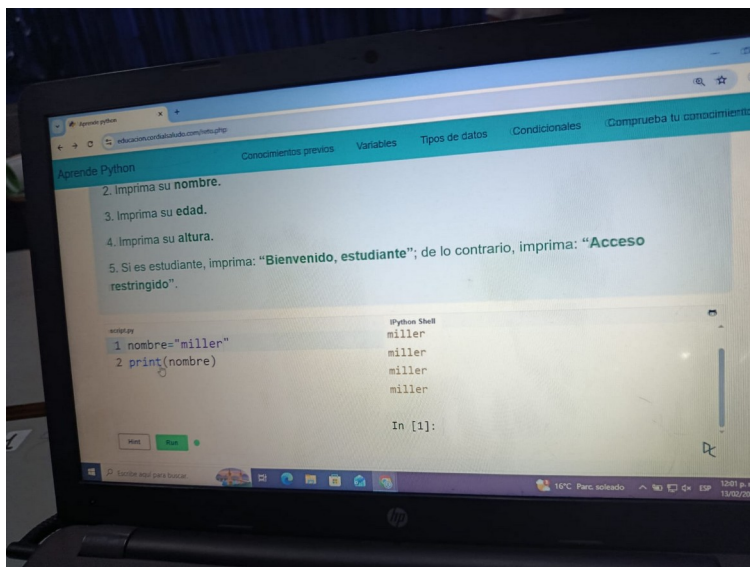
La estrategia de seguimiento de instrucciones se implementó mediante guías procedimentales apoyadas con texto, imágenes, Gifs y secuencias visuales que mostraban paso a paso la ejecución de tareas dentro del entorno de programación.

Durante las sesiones en el aula, se observó que los estudiantes utilizaban estos apoyos como referencia directa para replicar las acciones en el editor de código. La ejecución de las actividades se dio principalmente a través de la repetición de los pasos observados, lo que indica que el aprendizaje estuvo mediado por la reproducción de procedimientos más que por la comprensión autónoma.

Asimismo, se observó que los estudiantes recurrían a la observación del trabajo de sus compañeros para completar las actividades, lo que refuerza la idea de un aprendizaje basado en la imitación y la interacción social.

**Figura 29.**

*Ejecución del reto final usuario 20260213105359-7*

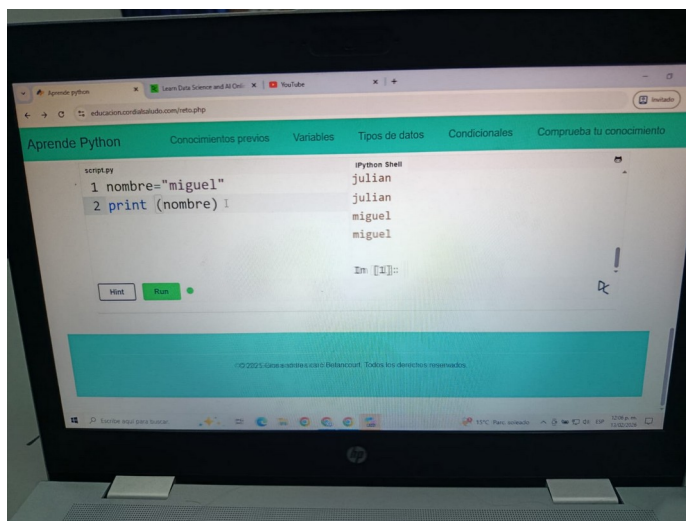


*Nota: Fuente propia*

Como se observa en la Figura 29, el estudiante con usuario 20260213105359-7, logra estructurar la primera parte del reto que es donde debe declarar una variable y asignarle un valor, en este caso, correspondía al nombre del estudiante, logró realizar la primera parte de escritura de código sin apoyo visual.

**Figura 30.**

*Ejecución del reto final usuario 20260213105914-1*

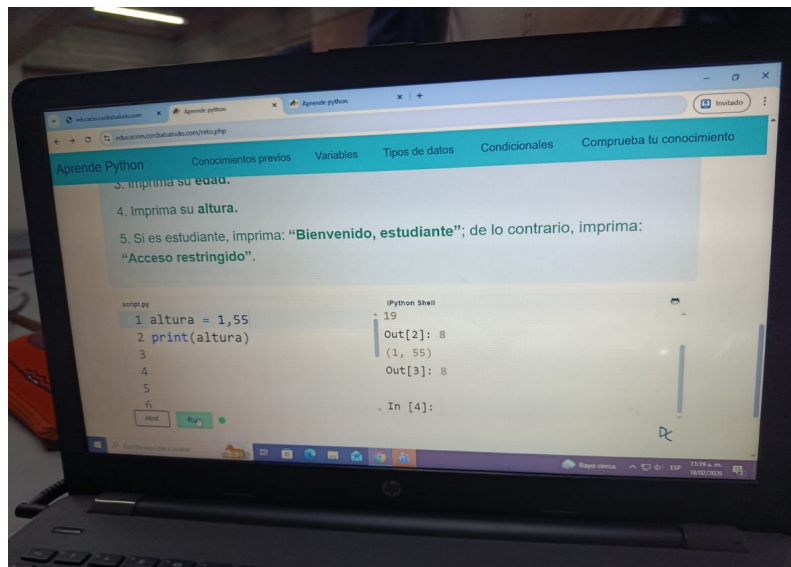


*Nota: Fuente propia*

Como se observa en la Figura 30, en el caso del estudiante 20260213105914-1, este logró llegar al mismo punto que su compañero. La diferencia es que, durante su proceso, primero observó qué hacía el otro para luego replicarlo en su editor de código.

**Figura 31.**

*Ejecución del reto final usuario 20260218104318-4*

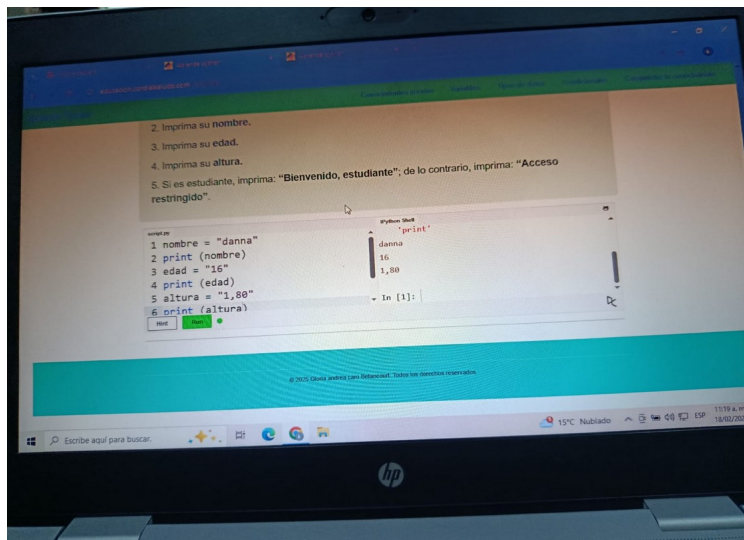


*Nota:* Fuente propia

En la Figura 31, se observa que el estudiante 20260218104318-4 inició con la declaración de la segunda variable que correspondía a la edad y debía asignar su propia edad y hacer todo seguido, y por esta razón borro la edad, pero se imprimió y solo cambió el nombre de la variable y su valor

**Figura 32.**

*Ejecución del reto final usuario 20260218104307-2*

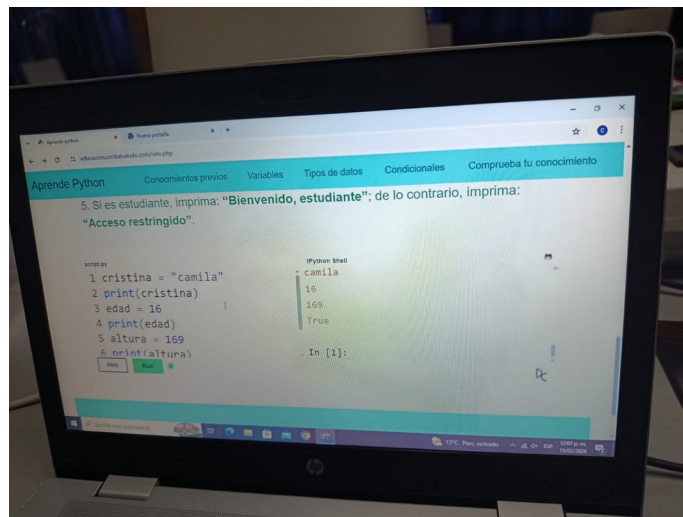


*Nota: Fuente propia*

En la Figura 32, se puede observar que el estudiante 20260218104307-2 logró avanzar un poco más y declarar tres de las variables que se les pedían en el reto final, aplicando así los conceptos de variables y tipos de datos, sin tener un apoyo visual, logró darle la estructura al código.

### **Figura 33.**

*Ejecución del reto final usuario 20260213105953-0*

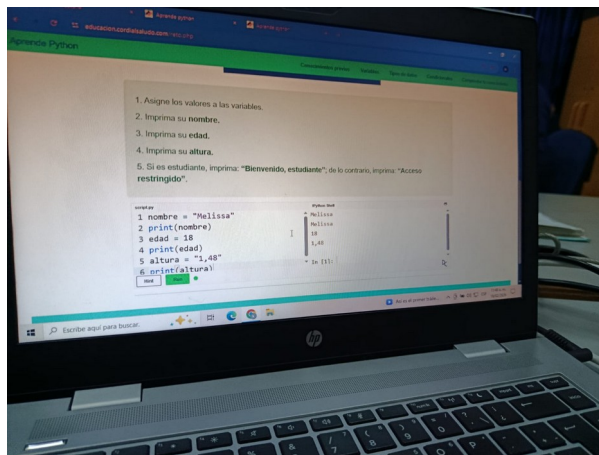


*Nota:* Fuente propia

En la Figura 33, se observa que se escribió una estructura de código por parte del estudiante 20260213105953-0 basándose en los ejercicios realizados en cada unidad temática, por eso, al finalizar asimila que lo que se le pidió fue una condicional de True o False y terminó poniendo lo que recordó de actividades realizadas anteriormente.

### **Figura 34.**

*Ejecución del reto final usuario 20260218112145-5*



*Nota:* Fuente propia

Por último, en la Figura 34 se observa que con este estudiante 20260218112145-5 se avanzó en tres de los requisitos que se pusieron utilizando variables y tipos de datos para resolver la estructura del código, no se terminó hasta llegar a la condicional solicitada, pero si las variables que declaró resultaron correctas.

Como se observa en las evidencias presentadas, los estudiantes lograron ejecutar parcialmente las tareas propuestas en el reto final. En la mayoría de los casos, se identificó la capacidad para declarar variables y asignar tipos de datos de manera correcta; sin embargo, persistieron dificultades significativas al intentar integrar estructuras de mayor complejidad lógica, como las condicionales.

Cuando los apoyos visuales eran limitados o estaban ausentes, los participantes tendieron a recurrir al ensayo y error o a la observación del trabajo de sus pares para completar las actividades. No obstante, el comportamiento principal no fue la exploración aleatoria, sino la reproducción de los pasos observados en sus compañeros o en ejercicios previos. En este sentido, los hallazgos indican que esta estrategia favorece la ejecución guiada y procedimental, pero presenta limitaciones, especialmente en tareas que demandan la aplicación independiente de la lógica de programación.

Con esto se pudo evidenciar que el aprendizaje mediante el seguimiento de instrucciones en esta población se dio principalmente a partir de la imitación de procedimientos. Si bien esta estrategia es efectiva para garantizar la culminación de tareas específicas, no garantiza por sí sola la asimilación de la lógica de programación, lo que la diferencia de las estrategias de comprensión y visualización.

## 6. Conclusiones

El presente estudio tuvo como propósito diseñar, implementar y evaluar un ambiente virtual de aprendizaje orientado a la enseñanza de conceptos básicos de programación en Python para estudiantes sordos, integrando estrategias pedagógicas de comprensión, visualización y seguimiento de instrucciones. A partir de los resultados obtenidos, se pueden formular las siguientes conclusiones:

1. Las dificultades para la comprensión y aplicación de conceptos básicos de programación en los estudiantes, formuladas por Mousley y Kelly (1998) pudieron evidenciarse claramente en los estudiantes que participaron en este estudio, siendo la impulsividad entendida como una tendencia marcada a avanzar demasiado rápido para resolver un problema sin detenerse a pensar; y las dificultades lingüísticas implicadas en el proceso de aprendizaje que involucra el uso de la Lengua de Señas Colombiana como lengua materna de los estudiantes sordos, el español como lenguaje en el que se desarrolla el proceso educativo en el aula y las palabras clave del lenguaje Python escritas en inglés, condición que dificulta la comprensión lectora y por ende, la comprensión de los problemas de programación y su resolución.

2. Se logró diseñar, implementar y evaluar un ambiente de aprendizaje virtual accesible para la enseñanza de programación en Python dirigido a estudiantes sordos. El ambiente integró estrategias probadas por investigaciones previas para facilitar el aprendizaje de las personas con discapacidad auditiva, las cuales fueron implementadas a través del diseño del ambiente de aprendizaje siguiendo tres estrategias: comprensión, visualización y seguimiento de instrucciones. Para su implementación se diseñaron recursos visuales, Gifs, imágenes y apoyos en Lengua de Señas Colombiana con el fin de facilitar la comprensión de los conceptos básicos de programación en Python a través de actividades accesibles para los estudiantes sordos.

Se observó que cuando las instrucciones estaban acompañadas de imágenes, gifs o apoyo en Lengua de Señas Colombiana, los estudiantes lograban comprender mejor las actividades y responder con mayor seguridad. Esto muestra que la accesibilidad y adaptación de recursos influyen directamente en los procesos de comprensión y aprendizaje de programación.

3. Se encontró que las estrategias de comprensión, visualización y seguimiento de instrucciones favorecieron el desarrollo de las actividades dentro del ambiente de aprendizaje. La estrategia de comprensión permitió que los estudiantes analizaran mejor las instrucciones antes de responder, la estrategia de visualización facilitó relacionar conceptos

abstractos con representaciones gráficas y el seguimiento de instrucciones ayudó a desarrollar las actividades paso a paso. Sin embargo, fue muy difícil lograr que los estudiantes avanzaran de forma más lenta a través de los contenidos, en pocas ocasiones se detuvieron a pensar. Esta falta de reflexión hizo que pasaran muy rápido por las diferentes secciones del entorno de aprendizaje. Esto sugiere que puede ser necesario que el escenario virtual se introduzca por lapsos muy breves durante muchas clases para tratar de que los estudiantes logren enfocarse en la información que se les presenta en cada sesión.

4. Aunque algunos conceptos presentaron mayores dificultades, especialmente las estructuras condicionales, el ambiente permitió generar espacios de aprendizaje más accesibles y acordes con las necesidades comunicativas de la población sorda. Se observó una mejora general en el posttest frente a los datos del pretest, especialmente en conceptos relacionados con variables y reconocimiento de instrucciones básicas. Aunque algunos estudiantes continuaron presentando dificultades en temas más complejos, los resultados mostraron avances en la comprensión de actividades. También se identificó que el aprendizaje estuvo acompañado constantemente por dinámicas de colaboración entre compañeros, observación de procedimientos y uso del ensayo y error como parte del proceso de resolución de problemas.

5. Con respecto a las decisiones técnicas implementadas dentro del ambiente virtual, se encontró que la organización de los contenidos siguiendo las tres estrategias de comprensión, visualización y seguimiento de instrucciones permite realizar una distribución secuencial de contenidos y el uso de interfaces simples que pueden facilitar la navegación y comprensión de las actividades. Esto puede aprovecharse mejor siguiendo un uso gradual del entorno en múltiples sesiones de clase.

6. Los recursos visuales ayudaron a mantener la atención de los estudiantes y apoyaron la comprensión de algunos conceptos básicos de programación, sin embargo, algunas actividades mostraron dificultades cuando las instrucciones escritas eran demasiado extensas. Esto demuestra que no basta únicamente con incorporar recursos visuales, sino que es necesario integrarlos de manera coherente con las actividades y objetivos de aprendizaje.

7. Finalmente, para la práctica docente, esta investigación demuestra la importancia de adaptar las estrategias de enseñanza a las necesidades comunicativas y visuales de los estudiantes sordos. El uso de recursos visuales, actividades organizadas paso a paso y apoyo en Lengua de Señas Colombiana favorece la comprensión y la participación dentro de las actividades. Además, se reconoce que la enseñanza de programación para la población sorda requiere procesos de mediación constantes, acompañamiento del docente y ambientes de

aprendizaje accesibles que reduzcan las barreras lingüísticas presentes en los contextos educativos.

### **6.1 Limitaciones y recomendaciones para futuras investigaciones**

Como limitaciones en el desarrollo de este trabajo de grado, se reconoce que el estudio se desarrolló con un grupo reducido de estudiantes y durante un tiempo corto, lo cual impidió profundizar en conceptos más complejos de la programación y realizar un seguimiento más prolongado del proceso de aprendizaje. Además, algunos estudiantes presentaban diferencias importantes en sus niveles de comprensión lectora y manejo tecnológico, lo que influyó en los resultados obtenidos.

Para futuras investigaciones se recomienda continuar desarrollando ambientes virtuales de aprendizaje accesibles dirigidos a la población sorda, incorporando más recursos en Lengua de Señas Colombiana, actividades interactivas y herramientas adaptativas que respondan a diferentes niveles de aprendizaje. También sería importante realizar otras investigaciones con grupos más amplios y durante periodos prolongados que permitan analizar el desarrollo del pensamiento computacional y el aprendizaje de programación a largo plazo.

## Referencias

Bransford, J. D., & Stein, B. S. (1984). *The IDEAL problem solver: A guide for improving thinking, learning, and creativity*. W. H. Freeman.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. En Proceedings of the 2012 Annual Meeting of the American Educational Research Association (p. 25).

CAST. (2021). *CAST Universal Design for Learning guidelines version 3.0*.

<https://udlguidelines.cast.org>

Celemín-Mora, J. C., & Flórez-Romero, A. (2018). *Percepciones sobre factores que inciden en los resultados de las pruebas Saber 11 de la población sorda: Una mirada desde tres instituciones educativas de Bogotá D.C., Colombia*. *Revista de la Facultad de Medicina*, 66(3), 349–356. <https://doi.org/10.15446/revfacmed.v66n3.61038>

Ministerio de Educación Nacional. (2017, 29 de agosto). *Decreto 1421 de 2017: Por el cual se reglamenta en el marco de la educación inclusiva la atención educativa a la población con discapacidad*. [mineduccion.gov.co](http://mineduccion.gov.co)

DataCamp. (s. f.). *DataCamp Light* (Versión 1.0) [Software de computación].

<https://github.com/datacamp/datacamp-light>

de Benito Crosetti, B., & Salinas Ibáñez, J. M. (2016). *La Investigación Basada en Diseño en Tecnología Educativa*. *RiiTE Revista Interuniversitaria de Investigación en Tecnología Educativa*, (1). <https://doi.org/10.6018/riite2016/260631>

Gehret, A. U., & Elliot, L. B. (2025). *Perceptions of e-learning by deaf and hard of hearing students using asynchronous multimedia tutorials*. *Educational Technology Research and Development*, 73, 1921–1949. <https://doi.org/10.1007/s11423-025-10476-z>

Hisyamuddin, M., & Tasir, Z. (2020). *An e-learning environment embedded with sign language videos: Research into its usability and the academic performance and learning patterns of deaf students*. *Educational Technology Research and Development*, 68(6), 2873–2911. doi.org

ICFES. (2021). *Guía de orientación Saber 11°. 2021, 1*.  
<https://www.icfes.gov.co/wp-content/uploads/2024/11/Guia-de-orientacion-Saber-11.%C2%B0-2021-1-Pdf-accesible.pdf>

Instituto Nacional para Sordos [INSOR]. (2020). *Educación bilingüe para sordos: Etapa escolar - Orientaciones pedagógicas*. Ministerio de Educación Nacional.

[https://educativo.insor.gov.co/wp-content/uploads/2020/12/Documento\\_01.pdf](https://educativo.insor.gov.co/wp-content/uploads/2020/12/Documento_01.pdf) (Trabajo original publicado en 2006).

Instituto Nacional para Sordos [INSOR]. (2022). *Lineamientos para la educación inclusiva de estudiantes sordos*. <https://www.insor.gov.co/>

Instituto Nacional para Sordos [INSOR], & Instituto Caro y Cuervo. (2006). *Diccionario básico de la lengua de señas colombiana* (1.<sup>a</sup> ed.).

<https://lenguasyliteraturasnativas.caroycuervo.gov.co/>

Kaewkamnerd, S., & Suwannarat, A. (2025). *Enhancing Computational Thinking of Deaf Students Using STEAM Approach*. *Education Sciences*, 15(5), 627.

<https://doi.org/10.3390/educsci15050627>

Mayer, R. E. (2001). *Multimedia learning*. Cambridge University Press.

McKenney, S., & Reeves, T. C. (2014). *Conducting educational design research*. Routledge.

Mousley, K., & Kelly, R. R. (1998). *Problem-solving strategies for teaching mathematics to deaf students*. *American Annals of the Deaf*, 143(4), 325–336.

<https://doi.org/10.1353/aad.2012.0082>

Severance, C. R. (2016). *Python para todos: Explorando la información con Python 3*. Edición independiente.

Shute, V., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking.

*Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>

UNESCO. (2020). *Global education monitoring report 2020: Inclusion and education: All means all*. <https://unesdoc.unesco.org/ark:/48223/pf0000373718>

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.). Harvard University Press.

## ANEXOS

### **Anexo A.** Acceso al ambiente virtual de aprendizaje (AVA)

El instrumento diagnóstico (pretest), las actividades desarrolladas y la evaluación final (postest) se encuentran integrados dentro del ambiente virtual de aprendizaje, específicamente en las secciones de “Conocimientos previos” y “Comprueba tu conocimiento”.

Disponible en: <https://educacion.cordialsaludo.com/?i=2>