Entornos Integrados de Desarrollo -IDEs- más utilizados por personas ciegas que desarrollan software mediante lectores de pantalla

Dayron Esteban Quevedo Velasco

Trabajo de grado para optar el título de Ingeniero de Sistemas

Director

Juan Francisco Mendoza Moreno

Doctor en Ingeniería Telemática

Universidad Santo Tomás Seccional Tunja

División de Arquitectura e Ingenierías

Ingeniería de Sistemas

2024

DEDICATORIA

Primero, dedico el éxito del presente proyecto a mi mamá, ya que con su esfuerzo y dedicación logró impulsarme para culminar con éxito mi carrera universitaria, también quiero incluir a los distintos docentes de la universidad, que en cada semestre infundieron en mí los conocimientos apropiados para terminar el programa académico. Por último, también le dedico el éxito de este proyecto a mis amigos y compañeros, ya que ellos también fueron parte fundamental en muchos momentos de este largo camino de esfuerzo y superación.

AGRADECIMIENTOS

Agradezco a los ingenieros con los que tuve la suerte de compartir en este proceso académico, especialmente a Carlos Andrés Guerrero, Juan Francisco Mendoza, Martha Contreras y a todos los docentes que influyeron en el éxito del proyecto, ya que sin su ayuda y dedicación no se podría llevarlo a su culminación.

TABLA DE CONTENIDO

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

RESUMEN

Este proyecto se centra en la accesibilidad de los entornos de desarrollo (IDEs) y lectores de pantalla para programadores personas ciegas. Su objetivo es identificar los IDEs más amigables para esta población y contribuir con una mayor inclusión en el ámbito de la programación.

La iniciativa analiza la evolución de la tecnología para personas ciegas en el desarrollo de software. Compara las características de accesibilidad de IDEs populares como Apache Netbeans, Microsoft Visual Studio Code y JetBrains PyCharm. Además, realiza encuestas a usuarios ciegos para comprender sus experiencias con estos IDEs.

El proyecto busca determinar los IDEs más accesibles para programadores ciegos y promover su inclusión en este campo. Su impacto potencial incluye la mejora de la accesibilidad de los IDEs para personas ciegas, la facilitación del aprendizaje y la práctica de la programación para este grupo, y la apertura de un mundo de oportunidades en el ámbito del desarrollo tecnológico.

ABSTRACT

This project explores the accessibility of Integrated Development Environments (IDEs) and screen readers for blind programmers. It aims to identify the most user-friendly IDEs for this population and contribute to greater inclusion in the programming world.

The initiative delves into the evolution of technology for blind software developers. It compares the accessibility features of popular IDEs like Apache Netbeans, Microsoft Visual Studio Code, and JetBrains PyCharm. Additionally, it conducts surveys with blind users to understand their experiences with these IDEs.

The project seeks to determine the most accessible IDEs for blind programmers and promote their inclusion in this field. Its potential impact includes enhancing IDE accessibility for blind individuals, facilitating learning and practicing programming for this group, and opening doors to a world of opportunities in the realm of technological development.

GLOSARIO

Accesibilidad: Se refiere a la capacidad de un entorno, producto o servicio para ser utilizado por todas las personas, independientemente de sus capacidades o discapacidades.

Discapacidad Visual: Incluye una gama de condiciones visuales que afectan la capacidad de una persona para ver, desde baja visión hasta ceguera total.

Entornos de Desarrollo Integrado (IDE): Aplicaciones de software que proporcionan herramientas completas para el desarrollo de software, incluyendo un editor de código, un compilador y un depurador.

Lectores de Pantalla: Programas de software que permiten a las personas con discapacidad visual usar computadoras mediante la conversión de texto en habla o Braille.

Introducción

En el presente proyecto, se aborda la problemática de la inclusión de personas con discapacidad visual en el ámbito de la educación superior en carreras relacionadas con la tecnología y la informática. Hoy, cada vez más personas desean estudiar estas carreras, incluyendo las personas con discapacidad visual que buscan desempeñarse laboralmente en áreas como la programación y el desarrollo web. Sin embargo, la culminación de las carreras universitarias por parte de esta población enfrenta diversos obstáculos significativos.

Uno de los principales problemas es la falta de capacitación de los docentes en la educación superior para orientar a estudiantes con discapacidad visual. Esta situación se debe tanto a la desinformación como a la falta de personal capacitado en la enseñanza del manejo de lectores de pantalla, afectando tanto a estudiantes como a docentes. Además, los entornos de desarrollo integrado (IDE) utilizados en estas carreras no brindan la accesibilidad necesaria para suplir las carencias visuales de los estudiantes. Entre los problemas de accesibilidad más comunes se encuentran la falta de señales sonoras en las líneas de código con errores, la ausencia de lectura de la compilación del código mediante lectores de pantalla y la dificultad en la interactividad con las consolas de comandos.

Estos inconvenientes, sumados a la falta de empatía en algunos casos, generan una brecha amplia entre los estudiantes con discapacidad visual y sus compañeros.

Con estas barreras, el proyecto busca apoyar a los docentes de educación superior con discapacidad visual en informática. Para ello, se realizará un análisis de la accesibilidad de los lectores de pantalla en conjunto con los entornos de desarrollo integrado. El objetivo es

determinar cuáles son los más apropiados para la enseñanza de esta población y proporcionar una mejor orientación a los docentes en la ejecución de planes de estudio accesibles.

El proyecto se estructura en varias fases. Primero, se hará una revisión bibliográfica sobre la evolución de los lectores de pantalla y los entornos de desarrollo integrado para accesibilidad. Posteriormente, se desarrollará una encuesta para determinar la experiencia del usuario utilizando estos entornos de desarrollo. El análisis se basará en criterios específicos como la generación de audio en la detección de errores, la presentación del resultado final del código, la accesibilidad en el desplazamiento por los menús del entorno de desarrollo y la ejecución de comandos en la consola.

Mediante este análisis, se identificarán los entornos de desarrollo más accesibles para personas con discapacidad visual y se establecerán lineamientos para los docentes de instituciones de educación superior. Además, se revisará la evolución de los lectores de pantalla para determinar los ajustes de accesibilidad aún necesarios. Finalmente, se propondrán lineamientos que permitan a las personas con discapacidad visual estudiar áreas de informática con mayor facilidad, contribuyendo así a una programación más accesible para todos.

1. Entornos Integrados de Desarrollo -IDEs- más utilizados por personas ciegas que desarrollan software mediante lectores de pantalla

1.1 Planteamiento del Problema

Cada vez son más las personas que desean estudiar carreras relacionadas con la tecnología y la informática. Lo anterior también es aplicable a las personas con discapacidad visual quienes también pueden buscar desempeñarse laboralmente en áreas como programación y desarrollo web.

Sin embargo, en la mayoría de los casos no se logra la culminación de las carreras universitarias por parte de esta población debido a varios factores: El primero de ellos es la falta de capacitaciones dirigidas a los docentes en la educación superior para orientar a los estudiantes con discapacidad. Esto se debe a la desinformación y en otros casos a la falta de personal capacitado en la enseñanza del manejo de lectores de pantalla, tanto a los estudiantes como a los docentes.

El otro gran inconveniente que se presenta a la hora de estudiar este tipo de carreras por parte de las personas en discapacidad visual es que los entornos de desarrollo integrado que se deben utilizar no brindan la suficiente accesibilidad para suplir la carencia visual. Dentro de los principales problemas de accesibilidad que presentan estos programas son:

- No emiten una señal sonora en las líneas de código que generan errores. Esto limita mucho que las personas en discapacidad realicen bien sus labores en la programación, ya que no se puede identificar claramente la ubicación de los errores en el código y las sugerencias de corrección.
- Falta la lectura mediante el lector de pantalla de la compilación del código.

- Dificulta la interactividad con las consolas de comandos.

Todos los inconvenientes mencionados, sumados a la falta de empatía que se presenta en ocasiones, generan una brecha amplia entre esta población y el resto de los compañeros del aula de clase.

1.2 Justificación

Considerando las significativas barreras que enfrentan los estudiantes con discapacidad visual en programas de informática en educación superior, tanto a nivel tecnológico como profesional, es imperativo brindarles un apoyo adecuado. Esto implica realizar un análisis exhaustivo de la accesibilidad de los lectores de pantalla y los Entornos de Desarrollo Integrado (IDEs). El objetivo principal es identificar los IDEs que mejor faciliten la enseñanza y el aprendizaje, asegurando que los estudiantes con discapacidad visual puedan interactuar de manera efectiva y productiva en el entorno de programación.

La importancia de este proyecto se fundamenta en abordar las barreras tecnológicas que limitan la participación plena de los estudiantes ciegos en el campo de la informática. Según datos recientes, las tasas de desempleo entre las personas ciegas son significativamente más altas en comparación con la población general. “De hecho, estudios indican que aproximadamente el 20% de las personas ciegas en edad laboral enfrentan dificultades para encontrar empleo adecuado” (Blindness, 2023), lo cual destaca la urgencia de mejorar las oportunidades educativas y profesionales para este grupo demográfico.

Mejorar la accesibilidad de los IDEs y los lectores de pantalla no solo incrementará la inclusión y la equidad educativa, sino que también promoverá un entorno donde todos los

estudiantes puedan desarrollar sus habilidades de programación de manera efectiva, independientemente de sus capacidades visuales.

Además, es fundamental reconocer que la industria de la tecnología y el desarrollo de software se beneficia enormemente de la diversidad de talentos y perspectivas. Capacitar a los estudiantes con discapacidad visual en el uso eficaz de herramientas accesibles no solo abre oportunidades educativas, sino que también enriquece la innovación y la creatividad en el campo tecnológico.

En resumen, este proyecto no solo busca identificar las mejores prácticas en accesibilidad tecnológica, sino también promover un cambio significativo en la manera en que los entornos educativos y profesionales abordan la inclusión digital. Al hacerlo, se aspira a crear un futuro donde todos los individuos, sin importar sus habilidades visuales, puedan participar activamente y contribuir de manera significativa en el campo de la informática y la tecnología.

1.3 Delimitación del proyecto

Al realizar este proyecto se pretende analizar la experiencia de los usuarios que usan lectores de pantalla para desarrollar software, para conocer cuáles son los lectores de pantalla más utilizados para programar y determinar cuáles son los entornos de desarrollo más accesibles con dichos lectores.

Con el documento se pretende que una gran cantidad de profesionales en las áreas de informática e ingeniería, puedan iniciar a generar aplicaciones más accesibles para las personas con discapacidad visual. Y que programadores y lectores conozcan en qué aspectos se debe mejorar la interfaz gráfica de cada entorno de desarrollo.

El proyecto será de gran apoyo para los docentes de asignaturas relacionadas con la ingeniería y la informática, porque en él se encontrará información sobre cómo activar la accesibilidad en los entornos de desarrollo y que lector de pantalla sea el mejor adecuado para utilizar un software. También les permitirá a las personas con discapacidad visual conocer cuál entorno de desarrollo es el más apropiado según el lector de pantalla que utilizan, esto sin duda facilitará el proceso para el aprendizaje de los estudiantes.

Mediante los lineamientos establecidos en este documento las personas con discapacidad visual podrán enfocarse desde el inicio de su carrera, en un entorno de desarrollo que les permita, ser competitivos a nivel profesional y posteriormente en las nuevas tecnologías que se pueda requerir aprender dentro de un entorno laboral. Al seguir un estándar que permita conocer cuáles entornos de desarrollo brindan una buena accesibilidad con lectores de pantalla como JAWS o NVDA, se logrará evitar que se tenga que explorar todos los entornos de desarrollo del mercado, generando así complicaciones en el aprendizaje de esta población.

Para la realización del proyecto se definieron varios objetivos, estos permitieron establecer los lineamientos ya mencionados anteriormente. Primero se definió la población con la que se trabaja, se estableció que para participar en el proyecto deben tener discapacidad visual total, o sea, que la persona no debe contener ningún tipo de baja visión, porque ellos no requieren lectores de pantalla.

Los participantes en el proyecto serán mayores de edad y también deben tener conocimiento al menos de un lector de pantalla.

Los participantes en el proyecto no son mayores de 30 años, esto se debe a que se pretende que las personas que participen del proyecto sean jóvenes estudiantes de carreras preferiblemente relacionadas con la informática.

No será necesario que los participantes manejen perfectamente todos los entornos de desarrollo, pero es mejor que conozcan la programación básica, esto les permitirá tener una mejor experiencia al manejar los softwares creados para el proyecto.

Finalmente, se seleccionaron diez personas que cumplían los requisitos mencionados y se realizaron pequeños experimentos con software creado en los principales entornos de desarrollo, así los usuarios determinaron la accesibilidad de los lectores de pantalla y de los entornos de desarrollo. De esta forma se logró establecer unos requisitos mínimos de accesibilidad, que permitieron a las personas con discapacidad visual mejorar considerablemente su aprendizaje con el desarrollo web.

Es importante aclarar que el documento no se enfoca en los tipos de discapacidad visual, porque la baja visión no se considera ceguera avanzada y, por ende, no requiere lectores de pantalla de ningún tipo. Las necesidades de las personas con baja visión difieren de los requerimientos de aprendizaje de la población con discapacidad visual, por ende, las capacidades cognitivas de estas personas se estimulan de una manera distinta.

Dentro del documento tampoco se tuvo en cuenta la accesibilidad de lectores de pantalla o entornos de desarrollo con discapacidades distintas a la visual. También es importante aclarar que no se pretende estudiar la accesibilidad de todos los lectores de pantalla que existen, esto debido a que son varios los que actualmente se cuentan y no todos son utilizados por personas con discapacidad visual. Debido a ello, solo se tendrán en cuenta los más relevantes a lo largo de la historia.

Tampoco se estudiaron los entornos de desarrollo existentes, los seleccionados se eligieron con una consulta para determinar cuáles son los más utilizados por los programadores.

Finalmente, se aclara que el proyecto no pretende ser un estándar de accesibilidad que deben manejar los entornos de desarrollo, sino contribuir a investigaciones existentes y así mejorar la programación accesible.

1.4 Objetivos

Objetivo General

Analizar comparativamente la experiencia del usuario de los Entornos Integrados de Desarrollo -IDEs más utilizados, mediante lectores de pantalla para personas ciegas que desarrollan software, con el fin de determinar los IDEs más accesibles para esta población.

Objetivos Específicos

1. Identificar la evolución de los lectores de pantalla con respecto a la inclusión de personas ciegas dedicadas al desarrollo de software, mediante una revisión bibliográfica, con el fin de determinar condiciones de accesibilidad en estos dispositivos.

2. Analizar la evolución de Apache Netbeans, Microsoft Visual Studio Code, Microsoft Visual Studio 2022 y JetBrains Pycharm, por medio de un estudio comparativo de las características de accesibilidad de las últimas tres versiones de estos IDEs.

3. Identificar la experiencia del usuario de personas ciegas con el lector de pantalla Freedom Scientific Jaws, mediante un cuestionario personalizado según los IDEs mencionados, dirigido a personas en esta condición.

4. Especificar los requisitos mínimos deseables de los lectores de pantalla y de los IDEs mencionados para desarrollar software por parte de personas ciegas, mediante el estándar IEEE-830.

2. Marco Referencial

2.1 Marco Teórico

El marco teórico del proyecto se centra en la accesibilidad en la educación superior para personas con discapacidad visual, específicamente en el contexto de las carreras de informática. Para comprender esta problemática, se consideran diversas teorías y estudios previos que abordan la inclusión educativa, la tecnología asistiva y las metodologías de enseñanza adaptativas.

Inclusión Educativa

La inclusión educativa se refiere a la integración de todos los estudiantes, independientemente de sus capacidades o discapacidades, en el sistema educativo general. Según la UNESCO (2005), la educación inclusiva es un enfoque que busca responder a la diversidad de necesidades de todos los aprendices a través de una mayor participación en el aprendizaje, las culturas y las comunidades, y reducir la exclusión en la educación. La inclusión de estudiantes con discapacidad visual en programas de informática presenta desafíos específicos que deben ser abordados para garantizar una educación equitativa y de calidad.

Tecnología Asistiva

La tecnología asistiva se refiere a cualquier herramienta, equipo, software o sistema utilizado para aumentar, mantener o mejorar las capacidades funcionales de las personas con discapacidades. Los lectores de pantalla y los entornos de desarrollo integrados accesibles son ejemplos de tecnología asistiva crucial para los estudiantes con discapacidad visual. Estudios

como los de Kelly (2011) han demostrado que el uso de tecnología asistiva mejora significativamente el rendimiento académico y la participación de estos estudiantes en entornos educativos.

Metodologías de enseñanzas adaptativas

Las metodologías de enseñanza adaptativas se centran en adaptar las estrategias de enseñanza para satisfacer las necesidades individuales de los estudiantes. En el contexto de la educación superior para personas con discapacidad visual, esto implica la capacitación de los docentes en el uso de tecnologías asistivas y la adaptación de los materiales de enseñanza. El modelo de Diseño Universal para el Aprendizaje (DUA) propone un marco para crear ambientes de aprendizaje inclusivos que consideren las diversas necesidades de los estudiantes desde el principio (CAST, 2011).

Dentro de la siguiente sección se podrá encontrar el desarrollo del primer objetivo, en el cual se detalla a profundidad la evolución histórica de los principales lectores de pantalla utilizados por personas con discapacidad visual, se encontraron los lectores de pantalla organizados en orden cronológico, con el fin de facilitar al lector la comprensión del tema y su constante crecimiento a lo largo de la historia.

2.2 Marco Conceptual

Lectores de Pantalla

Los lectores de pantalla son programas de software que convierten información visual en información audible o táctil, permitiendo a personas ciegas o con baja visión interactuar con dispositivos digitales, incluyendo computadores y dispositivos móviles.

Componentes Principales.

- Captura de Información: “el lector de pantalla accede al contenido visual mediante técnicas como APIs de accesibilidad (por ejemplo, Accessibility APIs en Windows o UI Automation), OCR (Reconocimiento Óptico de Caracteres) para texto no accesible directamente, o técnicas de intercepción de pantalla” (María Cholakova, 2023).
- Procesamiento de Información: “la información capturada se procesa y se transforma en una forma perceptible para el usuario, utilizando síntesis de voz, Braille dinámico, o combinaciones de ambas según las preferencias del usuario” (accessibilityspark.com, 2023).
- Interacción con el Usuario: “los lectores de pantalla proporcionan formas de navegación por la interfaz de usuario, permitiendo a los usuarios ciegos explorar y manipular elementos visuales mediante comandos de teclado, gestos táctiles o dispositivos de entrada alternativos” (blazie.co.uk, 2024).

Funcionalidades Clave para Desarrollo de Software

- Navegación por Código: Facilitan la navegación eficiente por el código fuente, resaltando sintaxis y estructura mediante audio o Braille, lo cual es crucial para la comprensión y la depuración.

  > "Inspección de Elementos de Interfaz de Usuario (UI): Permite explorar y manipular componentes de la interfaz de usuario, como botones, cuadros de texto y menús, para asegurar la accesibilidad y usabilidad del software desarrollado" (Alesya Prytulenets, 2023).

- Depuración de Errores: "Ayuda a identificar y corregir errores mediante la lectura de mensajes de error, trazas de pila y variables en tiempo de ejecución" (alooba.com, 2023).
- Documentación y Ayuda Contextual: "Proporciona acceso a la documentación del código y a recursos de ayuda contextual, mejorando la eficiencia y la comprensión del desarrollo" (desku.io, 2022).

Consideraciones de Diseño y Desarrollo

- Accesibilidad desde la Etapa de Diseño: "Integrar principios de diseño accesible desde la concepción del software para facilitar la interoperabilidad con lectores de pantalla" (Olga Ervalde, 2024).
- Pruebas de Accesibilidad: Realizar pruebas exhaustivas con usuarios ciegos o con baja visión utilizando lectores de pantalla, para identificar y corregir barreras de accesibilidad.
- Compatibilidad con Tecnologías Asistivas: "Asegurar que el software sea compatible con una variedad de lectores de pantalla y dispositivos de entrada, incluyendo soporte para

estándares de accesibilidad como WCAG (Web Content Accessibility Guidelines)(moldstud.com, 2022).

Evolución y Mejoras Continuas

- Retroalimentación de Usuarios: Incorporar comentarios y sugerencias de usuarios ciegos que utilizan lectores de pantalla para mejorar la experiencia de desarrollo y la accesibilidad del software.
- Actualizaciones y Mantenimiento: Mantener actualizados los lectores de pantalla y el software desarrollado para adaptarse a nuevas tecnologías y necesidades de los usuarios.

Entornos de Desarrollo Integrado (IDE)

Los Entornos de Desarrollo Integrados (IDEs) son herramientas fundamentales para los desarrolladores de software, y adaptarlos para personas ciegas es crucial para facilitar su participación en el desarrollo tecnológico. A continuación, se detalla cómo los lectores de pantalla y las IDEs pueden interactuar:

Características Esenciales de IDEs Adaptados

- Navegación por Línea y Símbolo: los IDEs adaptados permiten la navegación eficiente mediante lectura de línea y símbolo, facilitando la orientación en el código fuente.
- Marcadores y Navegación Contextual: marcadores auditivos o táctiles y capacidades de navegación contextual ayudan a los desarrolladores ciegos a orientarse y navegar por grandes bases de código.

Adaptación de IDEs para Accesibilidad

- Personalización de Interfaz y Comandos:
  - Modos de Accesibilidad: “Modos específicos diseñados para aumentar el contraste visual, ajustar el tamaño del texto y reconfigurar la disposición de elementos para mejorar la accesibilidad” (growthaccelerationpartners.com, 2022).
  - Mapeo de Comandos y Atajos de Teclado: “Posibilidad de personalizar y mapear comandos y atajos de teclado según las preferencias y necesidades de los usuarios ciegos” (guarana-technologies.com, 2024).
- Compatibilidad con Tecnologías Asistivas:
  - Integración con Lectores de Pantalla: “Soporte completo para lectores de pantalla populares como JAWS, NVDA y VoiceOver, asegurando una experiencia de usuario fluida y accesible” (guarana-technologies.com, 2024).
  - Interfaz de Programación de Aplicaciones (API) de Accesibilidad: “Utilización de APIs estándar de accesibilidad para garantizar la compatibilidad con lectores de pantalla y otros dispositivos de asistencia” (guarana-technologies.com, 2024).

Impacto y Futuro

- Inclusión y Diversidad: Fomento de la inclusión de personas ciegas en la industria del desarrollo de software mediante herramientas accesibles y adaptadas.
- Avances Tecnológicos: Aprovechamiento de avances tecnológicos como inteligencia artificial y aprendizaje automático para mejorar la accesibilidad y funcionalidades de los IDEs adaptados.

Discapacidad Visual

Entendiendo la Discapacidad Visual

- Variedad de Necesidades: “La discapacidad visual abarca desde baja visión hasta ceguera total, cada una con necesidades específicas de acceso a la información y navegación en el entorno digital”(Maria Cholakova, 2023).
- Importancia de la Accesibilidad: “Adaptar herramientas como lectores de pantalla y IDEs es fundamental para asegurar que las personas con discapacidad visual puedan participar activamente en el desarrollo de software, aprovechando sus habilidades y conocimientos técnicos” (Michele Mcdonall, 2023).

Lectores de Pantalla y Acceso a la Información Visual

- Funcionamiento y Utilidad: “Los lectores de pantalla transforman información visual en formatos accesibles, como síntesis de voz o Braille dinámico, permitiendo a usuarios ciegos o con baja visión interactuar con interfaces gráficas y contenido digital” (Accesibilitysparck.com), 2023.

Inclusión y Empoderamiento en el Desarrollo de Software

- Impacto Positivo: Facilitar el acceso igualitario a herramientas tecnológicas promueve la inclusión y el empoderamiento de personas con discapacidad visual en la industria del desarrollo de software.

- Futuro y Avances Tecnológicos: Continuo avance en tecnologías como inteligencia artificial y aprendizaje automático para mejorar la accesibilidad y funcionalidades de las herramientas de desarrollo adaptadas.

Este marco conceptual integrado enfatiza cómo la adaptación de lectores de pantalla, IDEs y procesos de desarrollo de software puede mejorar significativamente la participación y contribución de personas con discapacidad visual en la creación de tecnología, promoviendo un entorno inclusivo y equitativo en la industria tecnológica.

3. Metodología

El proyecto está dividido en tres fases de desarrollo.

Primero, se realizó una investigación sobre la evolución de los lectores de pantalla, para determinar los cambios que estos programas han tenido en el tiempo.

Después se realizó la misma investigación, pero en esta ocasión con los entornos de desarrollo integrado, esto permitió determinar la accesibilidad de los lectores de pantalla.

Después de obtener los datos evolutivos respecto a los entornos de desarrollo integrado, esto permitirá conocer cuáles son los más utilizados por los desarrolladores. Y así se logró enfocar la encuesta de accesibilidad.

En la tercera fase de la metodología se realizó una encuesta a un grupo determinado de personas en discapacidad visual, mediante el cuestionario se pretende determinar la experiencia de ellos con los entornos de desarrollo más utilizados. Cada usuario tuvo interacción con un entorno de desarrollo distinto y utilizó un lector de pantalla, algunos trabajaron con Jaws y otros con NVDA.

Finalmente, se realizó un análisis de la encuesta para determinar las ventajas y desventajas de los usuarios al manejar dichos entornos de desarrollo, y se hicieron sugerencias para establecer lineamientos que permitan una mejor interacción entre desarrolladores, entornos de desarrollo y lectores de pantalla. Mediante estos lineamientos se inició la construcción de un camino para lograr en un futuro una programación más accesible para todos.

4. Resultados

4.1 Revisión bibliográfica

Lectores de pantalla

Para consultar adecuadamente sobre la evolución histórica de los lectores de pantalla, se hicieron búsquedas en distintas bases de datos académicas, para encontrar la mayor cantidad de información posible y documentar al lector de manera correcta con respecto a la evolución histórica de este tipo de software. Dentro de las revistas y bases de datos que fueron utilizadas se destacan: Scopus, JSTOR, PubMed y ProQuest.

La información obtenida en las bases de datos se estudió y posteriormente se resumió para documentarla en este proyecto.

Primeros lectores de pantalla

Con exactitud no se conoce el primer lector de pantalla que existió, sin embargo, se puede afirmar que en la década de los 60 inicia la creación de los primeros sintetizadores de voz para personas ciegas, el primer lector de pantalla del que se tiene conocimiento se llamó “The Reading Machine”. Este lector lo inventó Ruth Baldwin, aunque no era un lector de pantalla sofisticado, fue muy importante para que luego se inventaran los lectores de pantalla modernos. Este lector solo podía transcribir a voz textos sencillos y no mucho más, pero sin embargo fue importante para las personas con discapacidad visual de la época, ya que les permitió tener una alternativa distinta al Braille.

La motivación para crear el lector de pantalla surge debido a que su inventora Ruth Baldwin, tenía ceguera congénita, debido a ello vio la necesidad de implementar un lector de

pantalla que le permitiera dictar sus clases de inglés con una mayor facilidad, y así surgió la idea de crear The Reading Machine. El sistema funcionaba con sencillo, usaba foto detectores para escanear texto impreso y transformarlo en señales eléctricas, y finalmente esas señales se convertían en sonido.

Evolución de The Reading Machine

En la década de los 70 el dispositivo implementó terminales Braille, la idea era permitir al usuario leer utilizando dicho sistema o mediante el sintetizador de voz del lector de pantalla, si bien este lector tuvo mucho auge en las personas ciegas de la época fue sustituido en la década de 1980 por otros lectores de pantalla que tenían una mejor conversión de texto a voz, sin embargo, como ya se mencionó anteriormente, este lector fue el precursor de una gran evolución histórica que mejorarían significativamente la vida de las personas con discapacidad visual.

Lector de pantalla JAWS (1988)

Este lector de pantalla fue creado por la empresa Freedom Scientific, en sus inicios se limitaba a leer textos planos y permitía el manejo de funciones muy básicas, sin embargo, fue evolucionando rápidamente y en la década de los 90, implementó sintetizadores de voz de alta calidad, la idea era convertir la voz de máquina en voz humana, en sus inicios esto no fue posible en un 100%, pero con el transcurso del tiempo fue asemejando su voz a la de un humano. También cabe recalcar que en esa década JAWS implementó compatibilidad con

dispositivos Braille, permitiendo a los usuarios manipular información, ya fuera con la voz sintética del lector de pantalla o con un dispositivo táctil para leerlo. (Freedom Scientific,2024)

- Evolución y consolidación en el siglo XXI: JAWS en la década de los 2000 agregó compatibilidad con más sistemas operativos, dentro de estos se incluyó a Linux, y también a macOS, esto permitió un crecimiento significativo en el número de los usuarios del lector de pantalla. Ya que al incluir más sistemas operativos incrementa el número de personas que podían probar las funcionalidades de JAWS.
- Implementación de inteligencia artificial en el lector de pantalla JAWS: con el auge de la inteligencia artificial la compañía Freedom Scientific vio necesario implementar esta tecnología en su principal producto. Para realizar dicho proceso se tuvieron en cuenta varios aspectos, el primero de ellos fue conocer cuáles eran las necesidades principales de las personas con discapacidad visual, es decir cuáles eran los desafíos más grandes que enfrentan estas personas en el momento de manipular nuevas tecnologías utilizando el lector de pantalla JAWS.

Tras conocer esta información, la compañía inició a estudiar técnicas de IA, entre las que se incluyó el procesamiento del lenguaje natural (PLN), el aprendizaje automático y la visión artificial, para identificar las necesidades específicas de JAWS.

También se desarrollaron algoritmos personalizados para tareas como la predicción de texto y también para el reconocimiento de objetos en la pantalla.

Finalmente se integró la inteligencia artificial en JAWS, este proceso se realizó cuidadosamente para no abrumar a los usuarios con funciones complejas dentro del menú del lector de pantalla, sin embargo, se desarrollaron nuevos comandos para el manejo de JAWS, pero estos eran muy intuitivos y no causaron molestias en los usuarios del producto.

Las pruebas de la funcionalidad de la inteligencia artificial se hicieron con varios usuarios de JAWS, obteniendo sensaciones positivas de los usuarios, las actualizaciones del producto se realizan constantemente y con la ayuda de la inteligencia artificial tienen una gran experiencia de usuario. A continuación, se incluyen algunos ejemplos de las funciones que cumple la inteligencia artificial en JAWS:

- Predicción de texto: "JAWS utiliza la IA para predecir la siguiente palabra o frase que el usuario está escribiendo, mejorando la velocidad y la eficiencia de la escritura" (Freedom Scientific, 2024).
- Comprensión del contexto: "JAWS analiza el contexto de la conversación o el documento actual para proporcionar información más relevante y útil al usuario" (Freedom Scientific, 2024).
- Reconocimiento de objetos: "JAWS puede identificar objetos en la pantalla, como botones, enlaces y elementos de menú, y proporcionar descripciones o comandos contextuales" (Freedom Scientific, 2024).
- Control por voz: "JAWS permite a los usuarios controlar el lector de pantalla y las aplicaciones con comandos de voz naturales" (Freedom Scientific, 2024).

Impacto de la IA en el lector de pantalla JAWS.

La integración de la IA ha transformado a JAWS en un lector de pantalla más inteligente y adaptable, lo que permite a los usuarios ciegos o con baja visión acceder a la información y la tecnología de manera más eficiente, natural e intuitiva.

Eventos relevantes durante la historia de JAWS: la historia de este lector de pantalla cuenta con varios eventos interesantes desde su creación, es por ello que se mencionarán los más importantes dentro de su evolución continua:

- 1993: Se lanza JAWS para Windows.
- 1995: “Se integra el sintetizador de voz VocalEyes para una mayor calidad de sonido” (LoPresti, 2021).
- 1999: “Se introduce el soporte para Braille con BrailleLink” (LoPresti, 2021).
- 2001: Se lanza JAWS 8 con nuevas funciones de navegación y comandos personalizables.
- 2005: “JAWS 10 incorpora el motor de voz FreedomVoice para una experiencia de voz más natural” (LoPresti, 2021).
- 2012: JAWS 14 presenta la tecnología "Say and Go" para el control por voz y la integración con dispositivos móviles.
- 2018: “JAWS 18 introduce la tecnología "AI-Powered Narrator" para mejorar la comprensión del contexto y la predicción de texto” (LoPresti, 2021).
- 2023: JAWS 2023 ofrece compatibilidad con Windows 11 y nuevas funciones de accesibilidad para aplicaciones web y documentos PDF.

Impacto de JAWS en las personas ciegas.

Este lector de pantalla permitió un acceso de calidad a la información en línea por parte de personas ciegas y también acortó un poco más la brecha inclusiva para esta población, al día de hoy continúa siendo uno de los lectores de pantalla más utilizados por las personas, y si bien su costo puede ser un poco elevado vale la pena adquirir este producto para desempeñar

las actividades más importantes en el PC, tales como redactar documentos, realizar consultas en línea, ver videos de entretenimiento entre otros múltiples aspectos.

JAWS y los lenguajes de programación.

En la constante evolución del lector de pantalla JAWS no se quedó lado la accesibilidad de este lector de pantalla con los entornos de desarrollo integrado, JAWS tiene un gran soporte para muchos lenguajes de programación, entre los que destacan. JavaScript, Python, CSS, Java y muchos más, esto les permite a los usuarios navegar por el código identificando su sintaxis y también permite la corrección de errores y lectura de mensajes de advertencia.

Modos de lectura específicos para código en JAWS.

El lector de pantalla permite leer el código estructurado, en bloques, permitiendo así comprender cada línea. También cabe resaltar que JAWS lee las líneas de código con la entonación adecuada y también con el tono que se requiere.

Comandos de teclado específicos para programación.

JAWS proporciona una serie de comandos y atajos de teclado específicos para la programación que facilitan la navegación y la edición del código. Estos comandos permiten a los usuarios realizar tareas comunes como:

- Moverse entre líneas, funciones y bloques de código.
- Seleccionar texto y editarlo.
- Insertar y eliminar código.
- Ejecutar y depurar código.

- Acceder a información de contexto y documentación.

JAWS también está integrado con la mayoría de los entornos de desarrollo, lo que permite al usuario tener soporte para la configuración de accesibilidad de este software.

Ventajas y desventajas de utilizar JAWS.

Como todo producto tecnológico tiene ventajas y desventajas al usarlo, por eso se anexarán los principales beneficios que JAWS les ofrece a sus usuarios y también los posibles contratiempos que se pueden presentar al usar este lector de pantalla.

Ventajas de accesibilidad.

- Uso del ordenador para personas ciegas o con baja visión: "JAWS permite a las personas con discapacidades visuales utilizar ordenadores con Windows de forma independiente, convirtiendo la información en pantalla en voz sintética o en Braille" (Fundación Americana para Ciegos (AFB), 2023).
- Lectura de contenido variado: JAWS puede leer una amplia gama de contenido, incluyendo documentos de texto, páginas web, correos electrónicos, software y más.
- Navegación y control del sistema: "JAWS permite a los usuarios navegar por el sistema operativo Windows, utilizar aplicaciones y realizar tareas comunes con comandos de teclado y voz" (Fundación Americana para Ciegos, 2023).
- Compatibilidad: JAWS es compatible con una amplia gama de aplicaciones de software y hardware, lo que lo convierte en una herramienta versátil para usuarios con diferentes necesidades.

Ventajas de funcionalidad.

- Personalización: “JAWS ofrece una amplia gama de opciones de personalización para adaptar la experiencia del usuario a sus preferencias, incluyendo la voz, la velocidad de lectura, los atajos de teclado y más” (Highsmith, 2000).
- Funciones avanzadas: JAWS incluye funciones avanzadas como reconocimiento de voz, lectura de OCR, soporte para múltiples monitores y más.
- Actualizaciones constantes: JAWS se actualiza constantemente con nuevas funciones y mejoras para ofrecer la mejor experiencia posible a sus usuarios.

Desventajas de utilizar JAWS.

Una de las principales desventajas que tiene este lector de pantalla es su elevado costo, en la mayoría de países JAWS no es gratuito, esto genera que los usuarios deban pagar altos porcentajes de dinero por adquirir una licencia, sin embargo es importante aclarar que el software en Colombia es gratuito, esto se debe al convenio que realizó el gobierno nacional con la empresa Freedom Scientific, este convenio le permite a los usuarios utilizar el lector de pantalla de manera gratuita en el territorio nacional.

Desventajas en complejidad para utilizar el producto.

- Curva de aprendizaje: “JAWS puede tener una curva de aprendizaje para usuarios nuevos, debido a la gran cantidad de funciones y opciones disponibles” (Universidad de York, 2023).

- Compatibilidad con aplicaciones web: “La compatibilidad de JAWS con algunas aplicaciones web puede ser limitada, especialmente con aquellas que no están diseñadas adecuadamente para la accesibilidad” (Universidad de York, 2023).

En general, las ventajas de JAWS le permitieron convertirse en el lector de pantalla más utilizado a nivel mundial.

Lector de pantalla NVDA

Si bien el lector de pantalla JAWS es uno de los más utilizados no es el único del mercado, también existen otras alternativas las cuales también tienen muchos usuarios, una de ellas es el lector NVDA (NonVisual Desktop Access, 2024). “Fue creado en el año 2006 por Michael Courtois” (The Verge, 2022), un programador ciego, la idea surgió debido a que los lectores de pantalla existentes hasta la época como JAWS eran muy costosos, debido a esto, Michael decidió crear un lector de pantalla para el sistema operativo Windows, su idea era que NVDA sea de código abierto para que cualquier programador pueda modificarlo y también brindar aportes para lograr consolidar el lector de pantalla en el mercado. Dentro de los eventos y características principales de este lector se encuentran:

Primera versión pública de NVDA (2007).

La primera versión del software fue muy importante porque debido a ella se crearon los cimientos para futuras mejoras y consolidación entre las personas con discapacidad visual. Dentro de sus funciones principales se incluía lo siguiente.

La lectura de texto con cualquier síntesis de voz se incluyó desde la primera versión de NVDA, el lector de pantalla podía identificar cualquier tipo de carácter especial y todo tipo de texto que el usuario ingresara por teclado.

NVDA también contaba en su primera versión con comandos para navegar dentro de sus principales funciones, esto le permitía al usuario interactuar con este lector de pantalla de una manera adecuada y sencilla.

La accesibilidad del lector de pantalla también fue buena, ya que ofrecía compatibilidad con varias distribuciones del sistema operativo Windows, además contó con una documentación que permitía a los desarrolladores mejorar el código para futuras versiones.

Limitaciones de la versión 1.0 de NVDA.

Si bien la primera versión tuvo aspectos interesantes, no todo fue perfecto, existieron algunas limitantes que no permitieron que el lector de pantalla se expandiera rápido en el mercado, una de ellas fue la dificultad que experimentaban los usuarios con NVDA en el momento de realizar reconocimiento de objetos e imágenes en la pantalla, también cabe recalcar que con la primera versión de NVDA no se podía realizar una navegación profunda en Internet, habían páginas web que el lector de pantalla no podía detectar, también presentaba muchos inconvenientes para interactuar con software de terceros. Sin embargo, poco a poco estas limitaciones fueron mejorando y NVDA logró crecer rápido en el mercado de la tecnología accesible.

Evolución de NVDA.

Después de lanzar la versión 1.0 del lector de pantalla la evolución de este fue de manera acelerada y constante, esto se debió en gran medida a la acogida que tuvo el software por parte

de los usuarios. A continuación, se muestran los principales acontecimientos de la evolución de este lector de pantalla:

- 2010: “NVDA gana reconocimiento por su accesibilidad y facilidad de uso, atrayendo a una base de usuarios más amplia” (Equally.ai, 2022)
- 2011: “Se funda NV Access, una organización sin fines de lucro, para apoyar el desarrollo y la promoción de NVDA” (NVAccess, s/f).
- 2015: “NVDA se convierte en el lector de pantalla más popular para Windows, superando a las opciones comerciales” (NVAccess, s/f).
- 2018: “Se lanza la versión 2018 de NVDA, con mejoras significativas en el rendimiento, la compatibilidad y la usabilidad” (Equally.ai, 2022).
- 2021: “NVDA alcanza 1 millón de usuarios activos, consolidando su posición como líder en el mercado de lectores de pantalla” (NVAccess, s/f).

- Versión de NVDA 2018: esta nueva versión lanzada al mercado en diciembre de 2017 fue muy importante porque fue la que más cambios significativos trajo para la evolución del lector de pantalla, de hecho, esta versión es la que finalmente contribuye a que hoy en día sea líder del mercado en tecnología accesible. Dentro de las principales mejoras en el software se destacan las siguientes:
  - Rendimiento mejorado: “Se optimizó el código para mejorar la velocidad y la eficiencia, lo que resultó en una experiencia de usuario más fluida y receptiva”. (AudioEye, s/f).
  - Mayor compatibilidad: “Se amplió el soporte para una gama más amplia de software y hardware, incluyendo nuevas tecnologías como Windows 10 Fall Creators Update y pantallas braille modernas” (Equally.ai, 2022).

- Funciones de accesibilidad avanzadas: Se agregaron nuevas características para mejorar la accesibilidad en diversos escenarios, como: Reconocimiento de objetos en pantalla: Permite a los usuarios identificar y comprender elementos visuales, como imágenes, gráficos y tablas.
- Navegación web avanzada: "Facilita la navegación por páginas web complejas, incluyendo tablas, formularios y enlaces dinámicos" (Harvard University Accessibility Services, s/f).
- Soporte para Braille mejorado: "Se amplió el soporte para diferentes tipos de pantallas Braille y se agregaron nuevas funciones de Braille, como la edición en línea y la retroalimentación táctil" (Harvard University Accessibility Services, s/f).

La versión 2018 del software sigue siendo una de las más descargadas por los usuarios, lo que significa que realmente les ha dado un impacto positivo.

- La versión actual del lector de pantalla NVDA es la 2024, se estrenó en el mercado en marzo de este año y cuenta con las siguientes mejoras dentro del software.

  "Se mejoró la navegación por pestañas, permitiendo así cambiar entre pestañas utilizando comandos por teclado" (NVAccess, s/f).

  - También se mejoró el soporte en los formularios web, esto permitió a los usuarios rellenar campos de forma automática.
  - Mejoras en la interfaz de usuario.

Aunque NVDA es un lector de pantalla intuitivo, se mejoraron algunos aspectos de accesibilidad y de personalización, entre los que destacan estos. Las ventanas de diálogo de NVDA ahora tienen un diseño más intuitivo y accesible.

- Se ha mejorado la compatibilidad con las pantallas táctiles.

- Se han añadido nuevas opciones para personalizar la apariencia de las ventanas de diálogo.

La versión mejoró aspectos en los que los usuarios manifestaron inconformidad, pero aún faltan inconvenientes por solucionar, una de ellas es la falta de compatibilidad con versiones anteriores, esto genera que cuando se instala una nueva versión del software se tenga que configurar la personalización nuevamente. Sin embargo, este pequeño inconveniente no le quita el prestigio que este lector de pantalla ganó entre las personas con limitaciones visuales,

NVDA y los entornos de desarrollo.

NVDA también cuenta con funciones de accesibilidad para que los usuarios que deseen incursionar en la programación puedan hacerlo, ya que cada vez son más las personas con discapacidad visual que quieren dedicarse profesionalmente a programar sitios web, debido a ello, en cada versión nueva del lector de pantalla también se mejora la accesibilidad para este tipo de usuarios. A continuación, se citan sus principales funciones para desarrolladores:

- Lectura de código:
  - "NVDA puede leer código fuente en diferentes lenguajes de programación, como Python, Java, C++ y JavaScript" (Accessibility Developer Guide, s/f).
  - Ofrece diferentes modos de lectura, como la lectura línea a línea, por carácter o por sintaxis.
  - "Permite resaltar elementos del código, como variables, funciones y clases" (Accessibility Developer Guide, s/f).
  - Ofrece información contextual sobre el código, como la definición de variables y la documentación de funciones.

- Navegación por el código:
  - "NVDA permite navegar por el código usando atajos de teclado y comandos de voz" (Accessibility Developer Guide, s/f).
  - Se puede saltar entre diferentes elementos del código, como funciones, clases y métodos.
  - Se puede buscar texto dentro del código.
  - Se puede plegar y desplegar código para ocultar o mostrar partes específicas.
- Edición de código:
  - NVDA permite editar código usando el teclado o un teclado Braille.
  - "Ofrece diferentes modos de edición, como la inserción de texto, la eliminación de texto y el movimiento de texto" (Accessibility Developer Guide, s/f).
  - Permite completar código automáticamente.
  - Permite corregir errores de sintaxis.
- Depuración de código:
  - NVDA permite depurar código paso a paso.
  - Se puede establecer puntos de interrupción en el código.
  - Se puede examinar el valor de las variables durante la ejecución del código.
  - Se puede ver el historial de llamadas a funciones.
- Integración con herramientas de desarrollo:
  - NVDA se puede integrar con diferentes herramientas de desarrollo, como IDEs, editores de texto y depuradores.
  - Esto permite que NVDA proporcione información y funcionalidad específica para la herramienta de desarrollo que se esté utilizando.

- Complementos y scripts:
  - Hay una gran cantidad de complementos y scripts disponibles para NVDA que pueden ampliar su funcionalidad y adaptarla a las necesidades específicas de los desarrolladores. Estos complementos pueden proporcionar soporte para lenguajes de programación específicos, herramientas de desarrollo o flujos de trabajo de desarrollo.

Con estas funciones de accesibilidad mencionadas se evidencia que NVDA está pensado no solo para los usuarios que realizan actividades básicas en el PC, como redactar documentos o navegar en Google, también está creado para que los programadores ciegos puedan desempeñarse profesionalmente en las áreas de informática, también es por ello que este lector de pantalla es líder entre las personas ciegas que utilizan lectores de pantalla.

Ventajas y desventajas de NVDA.

En NVDA, como todos los lectores de pantalla, hay limitaciones que lo diferencian de sus competidores, pero también hace que los usuarios prefieran usar otro lector de pantalla. Por eso es necesario conocer sus ventajas y desventajas para elegir adecuadamente el lector de pantalla.

- Accesibilidad: NVDA es accesible para todo tipo de público, esto se debe a que es un lector de pantalla de código abierto y cualquier persona con conocimientos en programación puede optimizar el código para mejorar el software, además hay que recalcar que el lector de pantalla es gratuito lo cual le da un gran punto a su favor. También es muy accesible con pantallas Braille, permitiendo así a los usuarios elegir si desean leer mediante este sistema, o si prefieren utilizar la voz sintética. Su gama de

comandos es variada y amplia, esto permite utilizar el lector de pantalla con abreviaciones de teclado, logrando así una gran comodidad para el usuario final.

NVDA también permite a los usuarios realizar configuraciones personalizadas, y adaptar de esta forma su experiencia con el software, dentro de las principales funciones que pueden configurarse encontramos. Los atajos de teclado, la configuración de la voz, y también la presentación de la información.

Este lector de pantalla también es accesible con todas las versiones de Windows, lo cual no limita a los usuarios del producto a tener una versión del sistema operativo específica.

Desventajas de NVDA: como todo lector de pantalla también tiene limitaciones, a continuación, se muestra el listado de algunas de ellas.

- Curva de aprendizaje: "NVDA tiene una amplia gama de funciones y opciones, lo que puede requerir un tiempo de aprendizaje para familiarizarse con el software" (NVAccess, s/f).
- Algunos usuarios pueden encontrar que la interfaz es abrumadora al principio.
- "Dependencia del software: NVDA depende del sistema operativo Windows para funcionar. Esto significa que los usuarios no pueden usar NVDA en otras plataformas, como macOS o Linux" (Equally.ai, 2022).
- Problemas de compatibilidad: aunque NVDA es compatible con la mayoría de las aplicaciones y sitios web, puede tener problemas de compatibilidad con aplicaciones o sitios web específicos.

- Soporte técnico: al ser un software de código abierto, NVDA no cuenta con soporte técnico oficial de un proveedor. La asistencia depende de la comunidad de usuarios y desarrolladores.

En general, NVDA es una poderosa herramienta que puede brindar acceso a computadoras con Windows para personas ciegas o con baja visión. Si bien existen algunas desventajas a considerar, como la curva de aprendizaje y la dependencia del software, las ventajas de NVDA en términos de accesibilidad, flexibilidad y compatibilidad la convierten en una opción valiosa para muchos usuarios.

Conclusiones de NVDA.

El lector de pantalla sigue evolucionando para mejorar a sus usuarios, tanto en accesibilidad como en la interfaz de usuario, es uno de los mejores lectores de pantalla hoy.

Lector de pantalla para computadoras Mac.

Si bien los lectores de pantalla mencionados anteriormente son muy buenos y cuentan con millones de usuarios a nivel mundial, aún tiene varias limitantes, y una de ellas es que no están disponibles para todos los sistemas operativos, esto obligaba a todas las personas con discapacidad visual a tener que utilizar Windows. Debido a ello surge la necesidad de desarrollar un lector de pantalla para otros sistemas operativos, y el primero en incursionar en esto fue el sistema operativo Mac.

- Creación del lector de pantalla VoiceOver: en el año 2005 Mac inició a implementar en su sistema operativo un lector de pantalla con el fin de que fuese más accesible para personas con deficiencias visuales, pero con un enfoque mayor en la ceguera. “La primera versión de

este sistema operativo en implementar el lector de pantalla fue Mac OS X 10.4 Tiger" (American Foundation for the Blind, s/f), sin bien el VOICEOVER no tenía en ese momento las funcionalidades de un lector de pantalla consolidado como JAWS, esta versión sentó las bases para la accesibilidad en el sistema operativo Mac y permitiendo a los usuarios la exploración de otros sistemas operativos diferentes a Windows. Las principales funciones del lector de pantalla eran las siguientes:

- Lectura de texto: "VoiceOver podía leer el contenido de la pantalla en voz alta, incluyendo documentos, correos electrónicos, páginas web y aplicaciones" (American Foundation for the Blind, s/f).
- Navegación por teclado: Los usuarios podían navegar por la interfaz de usuario utilizando comandos de teclado, lo que les permitía acceder a menús, botones y otros elementos sin necesidad de usar un ratón.
- Soporte para Braille: "VoiceOver podía conectarse a dispositivos Braille para que los usuarios pudieran leer y escribir en Braille" "VoiceOver podía leer el contenido de la pantalla en voz alta, incluyendo documentos, correos electrónicos, páginas web y aplicaciones" (American Foundation for the Blind, s/f).
- Personalización: VoiceOver ofrecía opciones para personalizar la velocidad de lectura, la voz utilizada y otros aspectos de la experiencia de lectura.

Limitaciones.

- Precisión: La precisión de la lectura no era perfecta, especialmente con contenido complejo o formateado.

- Compatibilidad: “No todas las aplicaciones y sitios web eran compatibles con VoiceOver” (CSS-Tricks, 2022).
- Funciones avanzadas: Carecía de funciones avanzadas como el reconocimiento de imágenes o el control por voz.

Sin duda, esta versión de VOICEOVER era limitada, pero fue muy acogida por los usuarios de la época, que dieron el aval con sus comentarios positivos para que el lector de pantalla siguiera creciendo.

- Versión 2.0 del VOICEOVER: “en el año 2007 Mac OS X 10.5 Leopard, incluye mejoras en su lector de pantalla” (American Foundation for the Blind, s/f), esto con el fin de darle mayor crecimiento en el mercado y atraer a un mayor número de usuarios. Entre los cambios más significativos destacó la mejora de la lectura, ya que el lector de pantalla experimentaba imprecisiones al deletrear o leer las palabras, estos errores se corrigieron y se implementó la compatibilidad con dispositivos Braille, esto se debió a que sus competidores de Windows ya tenían esta función implementada.
- Crecimiento de VOICEOVER en el año 2009: “en el año 2009 en Mac OS X 10.6 Snow Leopard, se logró introducir una avanzada función, la cual llamaron "VoiceOver Utilita", su principal función fue mejorar aún más la manera que tenían los usuarios de leer información con el lector de pantalla” (AbilityNet, 2019). A continuación, se muestran las principales características de esta función:

- Personalización de voces: “VoiceOver Utilita permitía a los usuarios seleccionar diferentes voces para la lectura, incluyendo voces sintetizadas y voces grabadas” (AbilityNet, 2019).
- Ajuste de la velocidad de lectura: Los usuarios podían ajustar la velocidad de lectura a su preferencia, haciéndola más rápida o lenta según sus necesidades.
- Control del tono y el volumen: VoiceOver Utility permitía ajustar el tono y el volumen de la voz de lectura, personalizando aún más la experiencia auditiva.
- Creación de diccionarios personalizados: Los usuarios podían crear diccionarios personalizados para agregar palabras o frases que VoiceOver pronunciara de manera específica.
- Configuración de reglas de pronunciación: “VoiceOver Utility permitía establecer reglas de pronunciación para controlar cómo se pronunciaban ciertos tipos de contenido, como números, direcciones o URLs” (AbilityNet, 2019).
- Importación y exportación de configuraciones: Los usuarios podían importar y exportar sus configuraciones de VoiceOver Utility, lo que les permitía compartirlas con otros dispositivos o realizar copias de seguridad.

En la actualidad esta función del lector de pantalla continúa evolucionando, esto impulsó mucho más la accesibilidad del VOICEOVER y logró un impacto muy positivo en todos los usuarios.

- Año 2011: “durante este año se implementó en el VOICEOVER una nueva tecnología llamada "Rosetta Stone", su función era facilitar la lectura de contenido web complejo”

(American Foundation for the Blind, s/f), esta tecnología tuvo que ser implementada debido a que el lector de pantalla aún tenía algunas dificultades para realizar lectura de algunos caracteres especiales en páginas web, con esta novedad VOICEOVER solucionó este inconveniente. A continuación, se listan sus características principales:

- Análisis semántico: “Rosetta Stone analizaba el código HTML de las páginas web para identificar elementos como encabezados, párrafos, listas y tablas” (American Foundation for the Blind, s/f).
- Comprensión del contexto: “La tecnología utilizaba técnicas de procesamiento del lenguaje natural para comprender el contexto del contenido web, identificando relaciones entre diferentes elementos de la página” (American Foundation for the Blind, s/f).
- Lectura fluida: VoiceOver utilizaba la información obtenida por Rosetta Stone para leer el contenido web de manera fluida y natural, siguiendo la estructura lógica de la página y resaltando los elementos importantes.
- Soporte para contenido dinámico: “Rosetta Stone podía manejar contenido web dinámico, como páginas que se actualizan automáticamente o utilizan JavaScript” (Evoluted, 2022).

El impacto fue muy positivo, ya que permitió a los usuarios acceder a una gama de funciones en Internet a las que no era tan sencillo ingresar, tras implementarla se pudo navegar en todas las páginas web sin complejidad.

- Comandos de voz en el VOICEOVER: “en el año 2012 Mac OS X 10.8 Mountain Lion introduce "Dictation", esto con el fin de permitir a los usuarios controlar el lector de pantalla

mediante comandos de voz" (American Foundation for the Blind, s/f), además lo más interesante es que permite controlar también funciones como dictado por voz y otras funciones del PC.

- Año 2013: el lector de pantalla continúa evolucionando y para el año 2013 OS X Mavericks 10.9 logró la compatibilidad del lector de pantalla con otras funciones del sistema operativo Mac como lo fueron iBooks y email.
- Año 2016 hasta hoy: en 2016 se introdujo en el lector de pantalla una novedosa función que permitía a los usuarios trabajar a la vez remota, o sea, permite controlar otro ordenador con la versión más actualizada de VOICEOVER, esta innovadora función fue posible gracias a la versión del sistema operativo macOS Sierra 10.12. Los usuarios vieron con mucho agrado esto, ya que permite trabajar de manera remota con otra persona evitando desplazamientos y también ahorrando tiempo.
- VOICEOVER para dispositivos móviles: debido a la gran popularidad que el lector de pantalla adquirió en los usuarios a nivel mundial, se vió la necesidad de expandirlo no solo para computadores, también estaría disponible para dispositivos móviles con sistema operativo IPhone. "El primer dispositivo IPHONE en incluir el lector de pantalla fue el iPhone 3GS en el año 2009" (Apple Support, s/f). Al igual que en el computador la idea de incluir el lector de pantalla en iPhone fue generar mayor accesibilidad para todos, y además fue una estrategia de mercadeo para lograr una mejor comercialización de estos dispositivos entre las personas con discapacidad visual.
- Evolución del lector de pantalla: a continuación, se incluye un listado de los principales cambios que tuvo el VOICEOVER desde 2009 hasta la actualidad en los dispositivos móviles:

- 2011: “iOS 5 introduce "Rotor", un control giratorio virtual que permite a los usuarios navegar rápidamente por diferentes elementos de la pantalla” (Apple Support, s/f).
- 2012: iOS 6 incorpora "Zoom", una función que permite ampliar cualquier parte de la pantalla para una mejor visualización.
- 2013: “iOS 7 presenta "Speak Selection", una función que permite leer en voz alta cualquier texto seleccionado en la pantalla” (Apple Support, s/f).

- Consolidando la accesibilidad:
  - 2014: iOS 8 introduce "Screen Curtain", una función que oscurece la pantalla y resalta los elementos con el puntero, facilitando la navegación para usuarios (Apple Inc, s/f), con baja visión.
  - 2016: iOS 9 incorpora “Guided Access”, una función que permite restringir el uso del dispositivo a una aplicación específica y controlar la interacción con la pantalla.
  - 2017: “iOS 11 presenta "VoiceOver Recognition", una función que permite a los usuarios reconocer objetos y texto en imágenes utilizando la cámara del dispositivo” (Apple Inc., s/f).
  - 2018: “iOS 12 introduce "Speak Screen", una función que permite leer en voz alta el contenido de cualquier pantalla, incluyendo aplicaciones y páginas web” (Apple Inc., s/f).
  - 2019: iOS 13 incorpora "Sound Recognition", una función que identifica y anuncia sonidos ambientales.
  - 2021: iOS 14 presenta "Live Text", una función que permite leer texto directamente desde imágenes y videos.
  - 2022: “iOS 15 introduce "Focus Filter", una función que permite a los usuarios filtrar las notificaciones y llamadas entrantes para minimizar las distracciones” (Apple Inc., s/f).

- Conclusión del uso del lector de pantalla VOICEOVER en dispositivos móviles: VOICEOVER es uno de los principales lectores de pantalla hoy, gracias a su constante evolución, además de adaptarse a la moda de los dispositivos móviles llegando a un mayor número de adeptos, con el tiempo continuará implementando nuevas versiones y con mayores mejoras, para lograr la mayor accesibilidad posible.
- VOICEOVER para desarrolladores: al igual que los lectores de pantalla de Windows, VOICEOVER también es accesible para programar en los principales entornos de desarrollo. Esto se debe a que cada vez son más los programadores ciegos que además de utilizar Windows desean incursionar en otros sistemas operativos, así que VOICEOVER no podía quedarse atrás en funciones de accesibilidad para desarrolladores. A continuación, se listan las principales características que ofrece el lector de pantalla para programar de forma accesible.
- Entornos de desarrollo integrados (IDE):
  - Xcode: "El IDE de Apple para el desarrollo de software macOS, iOS, iPadOS, watchOS y tvOS, ofrece soporte completo para VoiceOver. Esto incluye la lectura de código fuente, la navegación por la interfaz de usuario, la depuración y la ejecución de código" (Apple Inc., s/f).
  - Visual Studio: "El IDE de Microsoft para el desarrollo de software Windows, ofrece soporte para VoiceOver a través de una extensión de terceros" (Perkins School for the Blind, s/f). Esta extensión permite a los usuarios leer código fuente, navegar por la interfaz de usuario y depurar código.
  - IntelliJ IDEA: "El IDE de JetBrains para el desarrollo de software Java, Kotlin, Scala y otros lenguajes, ofrece soporte para VoiceOver" (Perkins School for the Blind, s/f). Esto

incluye la lectura de código fuente, la navegación por la interfaz de usuario, la depuración y la ejecución de código.

  - Eclipse: “El IDE de código abierto para el desarrollo de software Java, C++, Python y otros lenguajes, ofrece soporte para VoiceOver a través de una extensión de terceros. Esta extensión permite a los usuarios leer código fuente, navegar por la interfaz de usuario y depurar código (Perkins School for the Blind, s/f).

- Los editores de texto son accesibles con los entornos de desarrollo más comunes, y este lector de pantalla permite trabajar con Sublime, uno de los editores más utilizados por los programadores, para usarlo se debe configurar en las opciones de Sublime Text la accesibilidad con lectores de pantalla. Este proceso también debe realizarse con Visual Studio Code. Uno de los aspectos más interesantes de VOICEOVER es que a diferencia de otros lectores de pantalla, este es muy accesible con herramientas de control de versiones, uno de los más utilizados es GIT, el cual ofrece soporte para VOICEOVER mediante consola de comandos y también utilizando el entorno gráfico.
- Aspectos importantes para los desarrolladores: el lector de pantalla, mediante una configuración especial, permite poner VOICEOVER en modo desarrollador. Esta función permite interactuar con el código de la mejor forma posible. A continuación, se listan las principales características del modo para desarrolladores:
  - Anunciar atributos: “Esta sección te permite especificar qué tipo de información adicional deseas que VoiceOver anuncie sobre elementos de la interfaz, como roles, estados y etiquetas” (Apple Inc., s/f).

- Verificadores de accesibilidad: Activa verificaciones para detectar posibles problemas de accesibilidad en las aplicaciones, como elementos sin etiquetas o con etiquetas incorrectas.
- Verbosidad: Controla la cantidad de detalles que VoiceOver proporciona al describir elementos y acciones.
- Gestos: Personaliza los gestos de VoiceOver para adaptarlos a tus preferencias.
- Salida de audio: Ajusta la configuración de sonido, como la velocidad de voz, el tono y el volumen.
- Registro de eventos: "Registra las interacciones de VoiceOver con la interfaz para un análisis posterior" (Apple Inc., s/f).

En conclusión, el VOICEOVER es un lector de pantalla también adecuado para los desarrolladores, trabaja bien con la mayoría de los entornos de desarrollo y continúa en constante evolución, esto permite que siga mejorando progresivamente. Con la llegada de VOICEOVER los desarrolladores lograron tener la posibilidad de elegir en que sistema operativo desean trabajar, esto por básico que sea en un principio parecía imposible, lo cual evidencia el compromiso de las grandes compañías que se encargan de desarrollar aplicaciones por mejorar la accesibilidad para las personas con discapacidad visual.

Ventajas y desventajas de utilizar VOICEOVER.

Este lector de pantalla tiene grandes beneficios para quienes lo utilizan, pero también algunos aspectos a considerar para instalar este tipo de asistencia visual, por eso es importante mencionar los aspectos a favor y encontrar del VOICEOVER.

A continuación, se listan las ventajas:

- Acceso a la información: “VoiceOver convierte el contenido visual en texto audible, permitiendo a los usuarios acceder a libros electrónicos, sitios web, documentos y más” (CSS-Tricks, 2022).
- Uso del dispositivo: Navegar por el sistema operativo, realizar llamadas, enviar mensajes de texto, utilizar aplicaciones y controlar funciones del dispositivo se vuelve posible gracias a las descripciones de voz y los comandos de VoiceOver.
- Independencia: Los usuarios con discapacidad visual pueden realizar tareas cotidianas de forma autónoma, sin depender de la asistencia de terceros.
- Productividad: VoiceOver ofrece atajos de teclado y gestos para optimizar la eficiencia en la realización de tareas.
- Accesibilidad universal: “La compatibilidad de VoiceOver con una amplia gama de aplicaciones y sitios web promueve la inclusión digital” (CSS-Tricks, 2022).

A continuación, se listan las desventajas:

- Curva de aprendizaje: “Familiarizarse con las funciones y comandos de VoiceOver puede requerir práctica y paciencia” (CSS-Tricks, 2022).
- Dependencia del dispositivo: VoiceOver funciona únicamente en dispositivos Apple, lo que limita su uso en otras plataformas.
- Precisión: En algunos casos, las descripciones de voz pueden ser inexactas o incompletas, especialmente con contenido complejo o dinámico.
- Compatibilidad: No todas las aplicaciones y sitios web están optimizados para VoiceOver, lo que puede generar dificultades de acceso.

- Privacidad: Compartir información personal a través de comandos de voz puede generar inquietudes sobre la privacidad.

- Conclusiones del lector de pantalla VOICEOVER: este lector de pantalla fue una gran iniciativa por parte de Apple, ya que pensaron en la posibilidad de expandir los lectores de pantalla a muchos más usuarios fuera del sistema operativo Windows, además también se expandió a los dispositivos móviles como el iPhone, esto permitió a las personas ciegas tener acceso a los teléfonos celulares de alta gama, mejorando así sus posibilidades de acceder a todo tipo de contenido en línea. Es uno de los mejores lectores de pantalla del mercado y sigue mejorando y expandiéndose a muchos más usuarios a nivel mundial.

Lector de pantalla para Linux.

Al igual que Windows, el sistema operativo Linux también cuenta con un lector de pantalla accesible para los usuarios que tienen discapacidad visual. Si bien existen varios lectores de pantalla para Linux el más utilizado en la actualidad es ORCA, este lector de pantalla fue creado en la década del 2000 por el Real Instituto Tecnológico de Melbourne (RIMT), la idea principal era brindar accesibilidad para las personas con discapacidad visual, esto debido a que en el momento no existía un lector de pantalla gratuito para los usuarios que deseaban utilizar el sistema operativo Linux.

Inicios del lector de pantalla.

Las primeras versiones de ORCA se caracterizaron por su enfoque modular y su arquitectura flexible, esto permitía agregar nuevas funcionalidades en el código con facilidad. Además, cabe recalcar que, desde las primeras versiones de este lector de pantalla ya tenía

compatibilidad para dispositivos Braille, esta funcionalidad causó gran impacto en Linux, sorprendidos por el nuevo lector de pantalla.

Durante los 2000 ORCA fue muy popular entre los usuarios por la inclusión de nuevas síntesis de voz agregadas, además recibió el apoyo de una comunidad de desarrolladores, que colaboraron para mejorar el lector de pantalla y convertirlo en el más utilizado por Linux, porque es el lector de pantalla instalado por defecto en la mayoría de las versiones de este sistema operativo.

- Versión 1.0 de ORCA: lanzada en el año 2002, la versión 1.0 del lector de pantalla contó con aspectos interesantes para su época, a continuación, se listan las principales características de este software.
  - Soporte para entornos de escritorio:
    - "Funcionalidad básica para navegar por el escritorio GNOME" (Oracle Corporation, s/f).
    - Interacción con elementos de la interfaz gráfica, como menús, barras de herramientas y ventanas.
    - Lectura de texto en ventanas y aplicaciones.
  - Síntesis de voz:

    "Motor de síntesis de voz SVOX para convertir texto en voz audible" (Oracle Corporation, s/f).
    - Opciones para elegir diferentes idiomas y voces.
    - Control de la velocidad y el volumen de la voz.
  - Soporte para Braille:
    - Interfaz con dispositivos Braille para la salida de texto y la entrada de comandos.

- Soporte para diferentes códigos Braille.
- Posibilidad de visualizar el texto en Braille y en la pantalla simultáneamente.

➢ Accesibilidad a aplicaciones:

- Lectura de contenido en aplicaciones básicas como editores de texto, navegadores web y clientes de correo electrónico.
- "Soporte para algunas tecnologías de accesibilidad como AT-SPI" (Oracle Corporation, s/f).

➢ Personalización:

- Opciones para configurar la voz, el Braille, la velocidad de lectura y otros aspectos del lector de pantalla.
- Creación de perfiles personalizados para diferentes entornos de uso.

➢ Facilidad de uso:

- Interfaz de usuario sencilla e intuitiva.
- Atajos de teclado personalizables para un acceso rápido a las funciones.
- Documentación completa y recursos de ayuda disponibles.

Si bien la primera versión del lector de pantalla no contaba con las mejoras que pueden tener las versiones más actualizadas, fue muy importante para su futura mejora y su posterior popularidad.

- Versión 2.0 del lector de pantalla ORCA: esta versión se lanzó al mercado en 2008 y presentó a los usuarios una mejoría considerable respecto a su versión anterior, y se listan las principales mejoras incluidas en esta versión.

➢ Síntesis de voz:

- "Integración del motor de síntesis de voz Speak, que ofrecía una calidad de voz superior y mayor compatibilidad con idiomas" (Oracle Corporation, s/f).
- "Implementación de técnicas de concatenación de fonemas para una pronunciación más natural y fluida" (Oracle Corporation, s/f).
- "Soporte para la lectura de texto con prosodia, transmitiendo mejor las emociones y el contexto del contenido" (Oracle Corporation, s/f).

- Soporte para Braille:
  - Ampliación del soporte para diferentes códigos Braille y dispositivos Braille.
  - Implementación de modos de Braille más avanzados, como el modo de línea completa y el modo de foco.
  - Posibilidad de personalizar la representación del texto en Braille, incluyendo la contracción y el espaciado.
- Accesibilidad a aplicaciones:
  - "Mejoras significativas en el soporte para aplicaciones web, incluyendo una mejor comprensión de la estructura del HTML y la lectura de contenido dinámico" (Oracle Corporation, s/f).
  - Mayor compatibilidad con una amplia gama de aplicaciones de escritorio, incluyendo software de oficina, herramientas de productividad y entornos de desarrollo.
  - "Integración con tecnologías de accesibilidad más robustas, como ATK y GATK" (Oracle Corporation, s/f).
- Características adicionales:

        - Implementación de un sistema de comandos más completo y flexible para controlar el lector de pantalla.
        - Introducción de nuevas funciones como la detección de obstáculos en la pantalla y la lectura de descripciones de imágenes.
        - Posibilidad de crear scripts personalizados para automatizar tareas y ampliar las funcionalidades de Orca.
    - Mejoras en la interfaz de usuario:
        - Interfaz de usuario más moderna y accesible, con soporte para temas y configuraciones personalizadas.
        - Diálogos de configuración más intuitivos y fáciles de usar.
        - Documentación y recursos de ayuda más completos y actualizados.
        - Con las mejoras incluidas en esta actualización el lector de pantalla ORCA se convirtió en el lector de pantalla oficial del sistema operativo Linux y sus versiones posteriores, además ganó mucha más popularidad entre los usuarios del producto.
- Versión 3.0 de ORCA y actualidad: esta nueva actualización se lanzó en 2018 y tuvo las siguientes mejoras dentro del software.
    - Integración con las últimas tecnologías:
        - "Soporte para las últimas tecnologías web y estándares de accesibilidad, como HTML5, ARIA y WAI-ARIA" (Bureau of Internet Accessibility, s/f)
        - Compatibilidad con aplicaciones web modernas y sitios web interactivos.
        - Mejor comprensión del contexto y la estructura de la información en la web.
    - Soporte para nuevas tecnologías de asistencia:

- "Integración con tecnologías de reconocimiento de voz como Vos y Rhvoice para el control por voz del lector de pantalla" (Bureau of Internet Accessibility, s/f).
- Soporte para pantallas táctiles y gestos multitáctiles para una interacción más intuitiva.
- "Compatibilidad con dispositivos de seguimiento de la cabeza y la mirada para el control del cursor y la selección de elementos" (Bureau of Internet Accessibility, s/f).

- Mayor personalización y opciones de configuración:
  - Interfaz de usuario renovada y moderna, con temas personalizables y opciones de diseño.
  - Configuraciones avanzadas para ajustar la voz sintética, el Braille, la velocidad de lectura, los sonidos y otros aspectos del lector de pantalla.
  - Posibilidad de crear perfiles personalizados para diferentes entornos de uso y necesidades individuales.
- Fortalecimiento de la comunidad de usuarios y desarrolladores:
  - "Documentación completa y actualizada disponible en línea y en formato Braille" (Bureau of Internet Accessibility, s/f).
  - Recursos de ayuda extensos, incluyendo tutoriales, guías y foros de soporte.
  - Comunidad activa de usuarios y desarrolladores que contribuyen al desarrollo y la mejora continua de Orca.
- Otras características notables:
  - Lectura de libros electrónicos en formatos EPUB y DAISY.
  - Soporte para podcasts y audiolibros.

- Detección y anuncio de eventos del sistema, como notificaciones y alarmas.
- Posibilidad de reproducir música y otros archivos de audio.
- Integración con asistentes virtuales como Alexa y Google Assistant.

Dentro de las mejoras más significativas que valoraron los usuarios se encuentra la accesibilidad web, esto especialmente para los programadores, ya que con esta actualización lograron iniciar su incursión con el HTML.

En la actualidad este lector de pantalla es muy sofisticado y cuenta con millones de usuarios en todo el mundo, gracias a ORCA las personas con discapacidad visual lograron acceder a un sistema operativo como Linux, el cual en la actualidad tiene mucha demanda en el ámbito laboral, por ello este lector de pantalla cada vez suma más adeptos en el mundo, ya que Linux está alcanzando e incluso superando en popularidad a Windows.

- ORCA para desarrolladores: este lector de pantalla cuenta con una notable accesibilidad para que los usuarios ciegos puedan escribir código y compilarlo en cualquier entorno de desarrollo o lenguaje de programación.

El lector de pantalla tiene documentación para trabajar en la mayoría de los lenguajes de programación populares, incluyendo: Java, Python, JavaScript, C++, entre otros.

Uno de los aspectos más relevantes de este lector de pantalla es que permite identificar las líneas más importantes del código, dentro de los cuales destacan: variables, funciones, comentarios, operadores entre muchos más.

Esto permite al usuario lograr comprender el código, línea a línea, y trabajarlo de manera adecuada. Este lector de pantalla tiene documentación completa, que permite al usuario usar comandos de teclado para manejar el código y el entorno de desarrollo.

Además, ORCA tiene un comando especial para identificar el nombre del archivo y en qué carpeta se encuentra ubicado, esto permite al usuario controlar de forma correcta la estructura de carpetas dentro de un proyecto.

Respecto a la depuración de código, Orca también es muy accesible y le permite al usuario identificar errores y también proporciona información sobre las variables. Pero lo relevante en este punto de la depuración es que Orca lee el resultado de la compilación del código, en un entorno gráfico o en una consola determinada, este aspecto se destaca porque no todos los lectores de pantalla leen el resultado final del código, sin duda es una gran ventaja para encontrar el sistema operativo y lector de pantalla para trabajar.

Orca tiene documentación para los editores de código más populares, entre los que destacan Visual Studio y Eclipse.

En conclusión, este lector de pantalla permite trabajar a los desarrolladores, brindando accesibilidad y genera comandos que permiten hacer de este lector de pantalla la primera opción para trabajar en Linux.

Ventajas y desventajas del lector de pantalla ORCA.

Este lector de pantalla, como todos los productos, tiene funciones que pueden ser útiles para los usuarios, pero también hay aspectos a tener al decidir usarlo. Aquí, se listan ventajas y desventajas de este lector de pantalla.

- Accesibilidad: ofrece una amplia gama de funciones para usuarios ciegos o con baja visión, incluyendo la lectura de texto, la navegación por el sistema y la interacción con aplicaciones.

- “Soporta una gran variedad de entornos de escritorio, lenguajes de programación y formatos de archivo” (GNOME Foundation, s/f).
- Altamente personalizable para adaptarse a las necesidades individuales de cada usuario.
- Gratuito y de código abierto: disponible de forma gratuita para su descarga y uso. El código fuente está abierto para su revisión y modificación por parte de la comunidad. Esto permite un desarrollo continuo y la incorporación de nuevas funciones por parte de la comunidad.
- Multiplataforma:
  - “Funciona en la mayoría de las distribuciones de Linux” (GNOME Foundation, s/f).
  - Compatible con dispositivos, incluyendo computadoras de escritorio, laptops y dispositivos móviles.
- Comunidad activa:
  - Cuenta con una comunidad grande y activa de usuarios y desarrolladores que brindan soporte y asistencia.
  - Hay muchos recursos en línea, como documentación, tutoriales y foros de soporte.
- Innovación:
  - Se encuentra en constante desarrollo, con nuevas funciones y mejoras que se implementan regularmente.
  - Incorpora las últimas tecnologías en accesibilidad para ofrecer la mejor experiencia posible a los usuarios.
- Integración: se integra con diversos entornos de escritorio, herramientas de desarrollo y asistentes virtuales. Esto permite una experiencia de usuario fluida y consistente en diferentes contextos.

- Desventajas:
- Curva de aprendizaje: puede requerir un poco de tiempo y esfuerzo para aprender a usarlo de manera eficiente. La gran cantidad de funciones y opciones puede ser abrumadora para algunos usuarios.
- Compatibilidad: puede no ser compatible con todas las aplicaciones y sitios web.
- Algunas aplicaciones web modernas o sitios web interactivos pueden no ser accesibles con Orca. No es compatible con Windows o macOS.
- Dependencia del software libre: requiere un sistema operativo Linux para funcionar.
- Soporte técnico:
  - Al ser de código abierto, el soporte técnico oficial puede ser limitado.
  - "La comunidad de usuarios suele ser la principal fuente de soporte" (GNOME Foundation, s/f).
  - Recursos limitados en algunos idiomas:
  - La calidad de la síntesis de voz y la disponibilidad de recursos pueden variar en algunos idiomas.

Orca es un buen lector de pantalla para trabajar en Linux, pero los aspectos mencionados se deben tener para decidir qué uso.

El lector de pantalla Orca permitió a los usuarios de Linux trabajar en este sistema operativo con lectores de pantalla, aunque faltan muchas cosas por mejorar, no se puede discutir si se sigue en un sistema operativo que parecía que nunca podría brindar una accesibilidad de calidad.

Conclusiones

A continuación, se muestra la tabla de conclusiones de cada lector de pantalla, con el fin de mostrar al lector final un resumen de los aspectos más importantes de cada uno de ellos en relación con los entornos de desarrollo.

Tabla 1. *Comparación entre lectores de pantalla analizados.*

Aspecto	JAWS	NVDA	VoiceOver	Orca
Accesibilidad con el entorno de desarrollo	Alta compatibilidad con diversos entornos de desarrollo como Visual Studio, PyCharm entre otros.	Alta compatibilidad con diversos entornos de desarrollo.	Buena compatibilidad con XCode y otros entornos de Apple.	Compatible con varios entornos de desarrollo en Linux.
Lectura de comandos en la consola	Lee los comandos y resultados en la consola con precisión.	Lee los comandos y resultados en la consola con precisión.	Lee los comandos y resultados en la consola en entornos macOS.	Lee los comandos y resultados en la consola en entornos Linux.
Comandos de teclado para utilizar el código	Amplia variedad de comandos de teclado personalizables.	Amplia variedad de comandos de teclado personalizables.	Soporte para comandos de teclado específicos de macOS.	Soporte para comandos de teclado específicos de Linux.
Comandos de teclado para navegar por los menús del código fuente	Soporte completo para navegación de menús con teclado.	Soporte completo para navegación de menús con teclado.	Excelente soporte para navegación de menús con teclado en macOS.	Buen soporte para navegación de menús con teclado en entornos Linux.

Aspecto	JAWS	NVDA	VoiceOver	Orca
Lectura de la compilación del código	Proporciona feedback claro sobre errores y resultados de la compilación.	Proporciona feedback claro sobre errores y resultados de la compilación.	Proporciona feedback claro sobre errores y resultados de la compilación en entornos Apple.	Proporciona feedback claro sobre errores y resultados de la compilación en entornos Linux.
Descripción de gráficos en las imágenes del entorno de desarrollo	Puede describir gráficos con plugins adicionales.	Puede describir gráficos con plugins adicionales.	Puede describir gráficos en aplicaciones compatibles con VoiceOver.	Capacidad limitada para describir gráficos, depende del entorno y las extensiones.

Estándar W3C

"Los estándares W3C, también conocidos como Recomendaciones W3C, son protocolos, pautas y tecnologías interoperables e internacionales creados por el Consorcio World Wide Web (W3C)." (TechTarget, s/f). Su objetivo es guiar la web hacia un mejor rendimiento y mejorar la funcionalidad operativa para navegadores y usuarios.

El W3C es una comunidad internacional donde organizaciones miembros, personal a tiempo completo y el público trabajan juntos para desarrollar estándares web. Estos estándares se basan en el consenso de la industria y se publican como Recomendaciones W3C después de un proceso de revisión exhaustivo.

Los estándares W3C son importantes por varias razones:

- “Promueven la interoperabilidad: Aseguran que los sitios web y las aplicaciones web funcionen en una variedad de navegadores, dispositivos y plataformas” (TechTarget, s/f).
- Mejoran la accesibilidad: Hacen que la web sea más accesible para personas con discapacidades.
- “Fomentan la innovación: Proporcionan una base sólida para el desarrollo de nuevas tecnologías web” (TechTarget, s/f).
- Protegen la privacidad del usuario: Ayudan a proteger la privacidad de los usuarios en línea.

Dentro de las clasificaciones de W3C existentes, destaca una, la cual se abordará con mayor profundidad dentro de este documento, hablamos de WCAG. Esta clasificación se encarga de establecer pautas para la accesibilidad web. Dentro de los aspectos que destaca este estándar se encuentra la accesibilidad web con lectores de pantalla.

- WCAG y las pautas de accesibilidad para personas con discapacidad visual: estas pautas fueron establecidas con el fin de brindar a las personas ciegas completa independencia en el momento de utilizar páginas web, dentro de ellas también hacen parte los entornos de desarrollo integrados, esto debido a que, aunque no son páginas web como tal si son software que se descargan en línea y funcionan como una aplicación más dentro del sistema operativo. Por ello, los entornos de desarrollo deben cumplir con las pautas establecidas en dicho estándar.

A continuación, se detallan las pautas más importantes establecidas para la accesibilidad web.

- Perceptible: “El contenido web debe ser perceptible para todos los usuarios, independientemente de sus limitaciones sensoriales”. (World Wide Web Consortium, 2018) Esto incluye:
  - Usar imágenes y gráficos con texto alternativo equivalente.
  - Estructurar el contenido de manera lógica y coherente.
  - Utilizar títulos y etiquetas descriptivas.

  En la perfectibilidad se debe incluir la descripción de cada imagen en los tutoriales de uso de entornos de desarrollo.

- Operable: “El contenido web debe ser operable para todos los usuarios, independientemente de sus habilidades motrices o cognitivas” (World Wide Web Consortium, 2018). Esto incluye:
  - Permitir que todo el contenido se pueda navegar y utilizar con un teclado.
  - Proporcionar suficiente tiempo a los usuarios para leer y comprender el contenido.
  - Evitar el uso de contenido que parpadee o se actualice automáticamente.
  - Permitir que los usuarios desactiven las funciones de autocompletar.
- Comprensible: “El contenido web debe ser comprensible para todos los usuarios, independientemente de su idioma o nivel de conocimiento” (World Wide Web Consortium, 2018). Esto incluye:

- Utilizar un lenguaje claro y sencillo.
- Evitar la jerga técnica.
- Proporcionar definiciones para términos técnicos.
- Organizar el contenido de manera lógica y coherente.

Dentro de este último aspecto caben varias aclaraciones:

La jerga técnica puede ser utilizada dentro de los entornos de desarrollo, ya que este tipo de software fueron creados para personas capacitadas en las áreas de programación, y las personas con discapacidad visual que utilizan este tipo de programas deberán tener el conocimiento apropiado para comprender la información. Esto no exime a los entornos de desarrollo de cumplir el último punto del estándar, donde se establece que el contenido debe ser claro y organizado y estructurado lógica y coherentemente.

- Robusto: El contenido web debe ser robusto y compatible con una amplia gama de tecnologías de asistencia. Esto incluye:
  - Usar código HTML válido y bien formado.
  - Proporcionar texto alternativo para todos los elementos gráficos.
  - Usar etiquetas descriptivas para los elementos de formulario.
  - Evitar el uso de contenido incompatible con las tecnologías de asistencia.

Niveles de accesibilidad de las páginas web y los entornos de desarrollo.

Las W3C define tres niveles para saber qué tan accesible son las páginas web, en el nivel A se encuentran los softwares que no cumplen con el estándar, en el nivel AA están las páginas

web con un nivel de accesibilidad intermedio, y en el nivel AAA tenemos los sitios web que brindan una accesibilidad muy alta.

Si bien estas pautas son muy importantes para brindar accesibilidad a las personas con discapacidad, lastimosamente los entornos de desarrollo integrados no están obligados, al menos de manera legal a cumplir estos requisitos, sin embargo, la mayoría de ellos cumple con este estándar al menos en su nivel intermedio y se esfuerzan por crear documentación accesible para todos.

En la siguiente sección se dará a conocer al lector un análisis descriptivo de las versiones más recientes de los siguientes entornos de desarrollo.

Visual Studio 2022, Visual Studio Code, NetBeans y PyCharm. Esto para determinar qué tan accesibles son estos entornos de desarrollo usando lectores de pantalla. El análisis se realizó con un estándar internacional (W3C), que se encargó de determinar los requisitos para tener una página web accesible y un software específico.

Dentro de los aspectos más relevantes que se encuentran dentro de dicho estándar se destacan los siguientes:

Perceptibilidad, comprensibilidad, operabilidad, y robustez. Cada concepto se abordará más adelante en este documento.

Con este análisis los lectores tendrán más idea de que los entornos de desarrollo integrados tan accesibles, además podrán tener más claridad de que aspectos faltan por mejorar en cada uno.

4.2 Análisis descriptivo de entornos de desarrollo y su accesibilidad.

Considerando el estándar explicado en el apartado anterior, se realizó el estudio comparativo de cuatro entornos de desarrollo aplicando los aspectos más importantes de la W3C. Para comprender mejor la forma en la que se realizó el estudio comparativo se realizó un pequeño programa en cada uno de los entornos de desarrollo estudiados, en NetBeans y PyCharm se desarrolló una pequeña calculadora básica mediante la cual se determinó si el entorno de desarrollo cumplía con los estándares de accesibilidad. Dentro de los aspectos para tener en cuenta destacan:

Comandos de teclado en el editor, lectura del código fuente con el lector de pantalla, organización estructurada de cada menú y entre otros aspectos que se verán reflejados en una tabla que muestra los resultados del estudio.

4.3 Instalación de los entornos de desarrollo.

Lo primero que se realizó fue instalar las versiones más actualizadas de cada software, para determinar cuánto han mejorado su accesibilidad desde sus inicios. A continuación, se muestran las figuras con los entornos de desarrollo instalados.

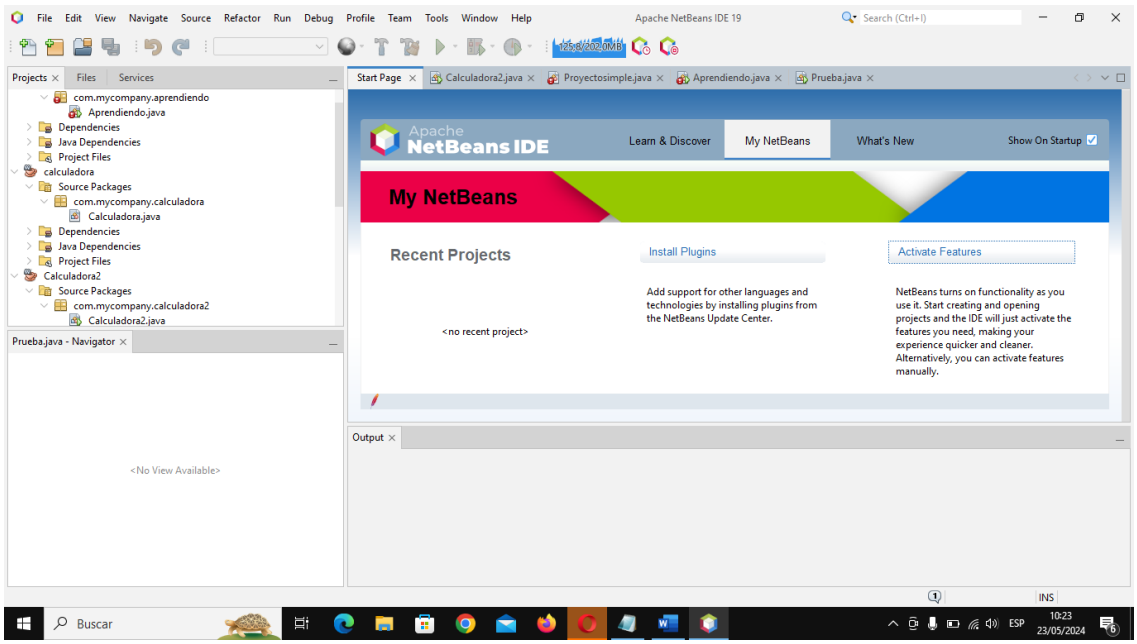

Figura 1. Pantalla inicial de NetBeans instalado

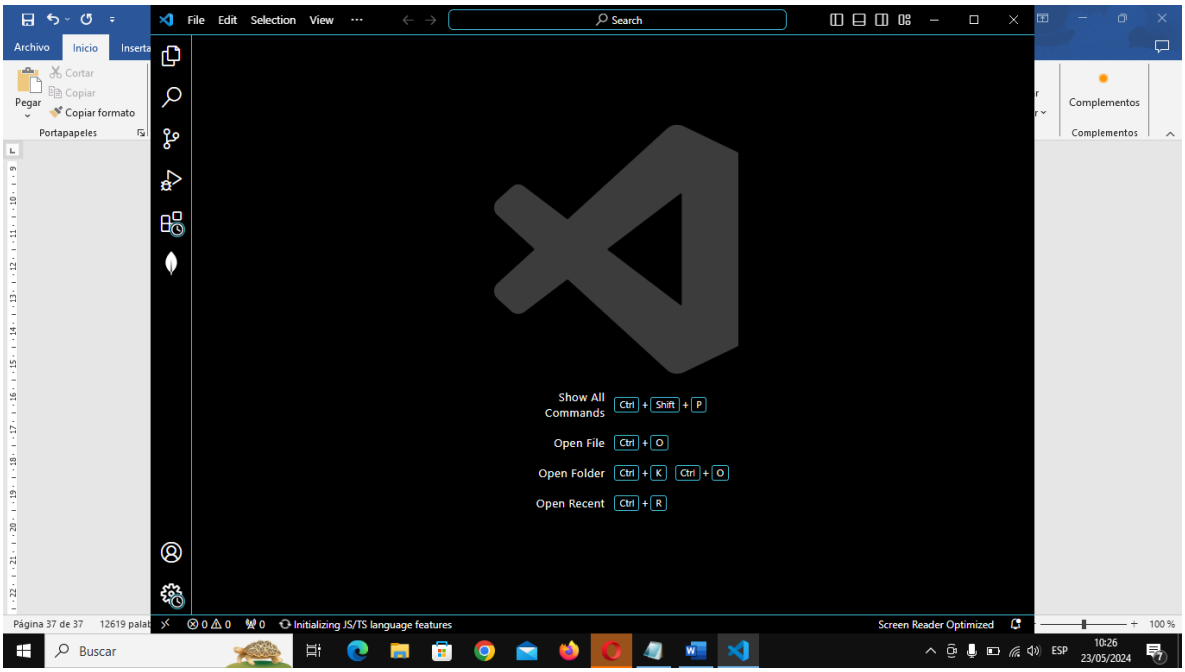

Figura 2. Pantalla inicial de Visual Studio Code instalado

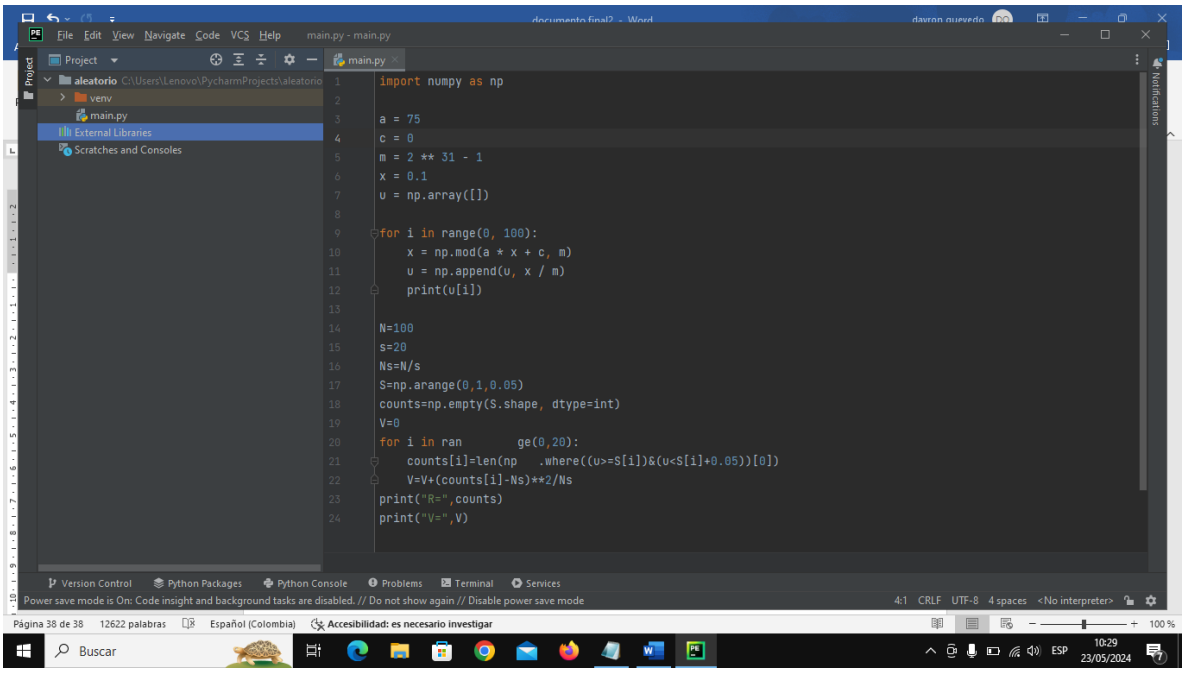

Figura 3. Pantalla de un proyecto con Pycharm instalado

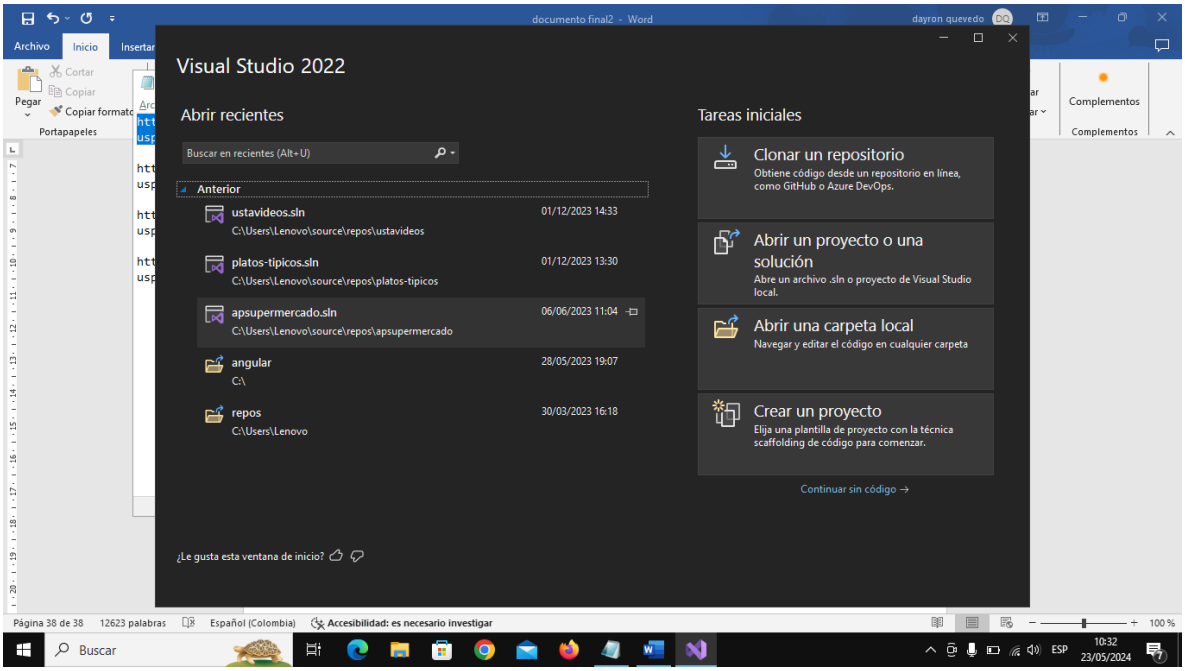

Figura 4. Pantalla inicial de Visual Studio 2022 instalado

4.4 Creación de las líneas de código.

Tras instalar cada entorno de desarrollo, se crearon los programas para realizar el estudio comparativo. En Visual Studio Code se desarrolló una página web con Angular y MongoDB, a esta página se le aplicó el estándar de accesibilidad explicado anteriormente, no solo a la página web, también al entorno Visual Studio Code.

En el entorno de desarrollo PYCHARM se desarrolló una calculadora, la cual se encarga de realizar las operaciones básicas (sumar, restar, multiplicar y dividir). La idea de la calculadora era permitir a los usuarios compilarla mediante consola y determinar la accesibilidad del entorno de desarrollo y la consola de comandos.

En Visual Studio 2022 se realizó un software usando el lenguaje de programación PHP, el software se encarga de registrar usuarios y enviar un correo a cada uno, igual que en los softwares anteriores, se permitió a los usuarios revisar la accesibilidad del entorno de desarrollo y manejar el teclado de la página.

En el entorno de desarrollo NETBEANS también se creó un programa básico que sumó 2 números y dio su resultado, el objetivo fue dar al usuario a explorar el código y los comandos del entorno de desarrollo.

A continuación, se muestran pequeños fragmentos del código realizado para evidenciar el trabajo realizado en cada entorno de desarrollo.

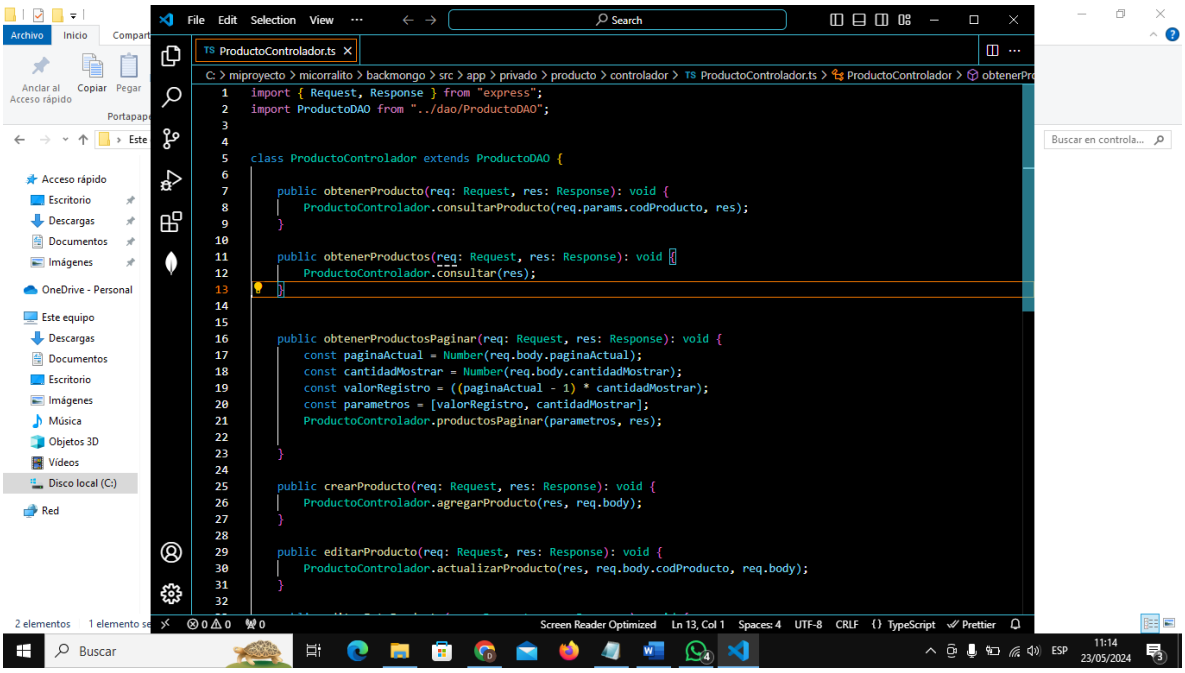

Figura 5. Código creado en Visual Code

El fragmento de código mostrado en la siguiente figura hace parte de la página web creada con Visual Studio Code.

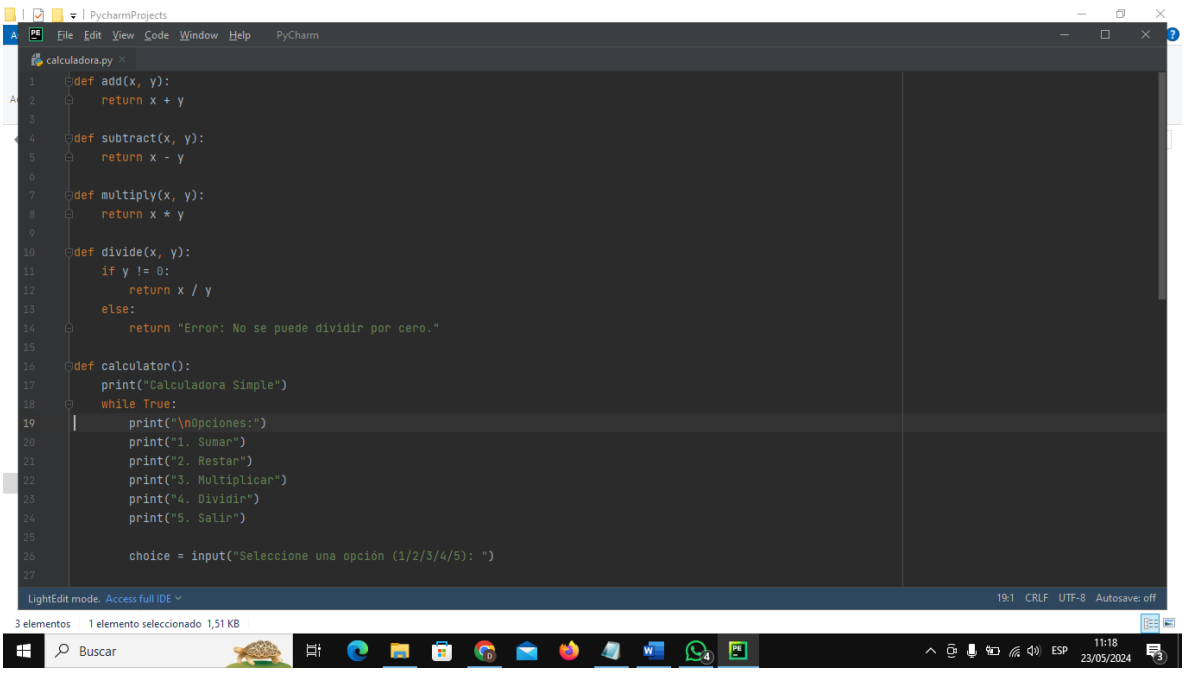

Figura 6. Código de página web en Visual Code

Este fragmento de código hace parte de la calculadora creada en PYCHARM.

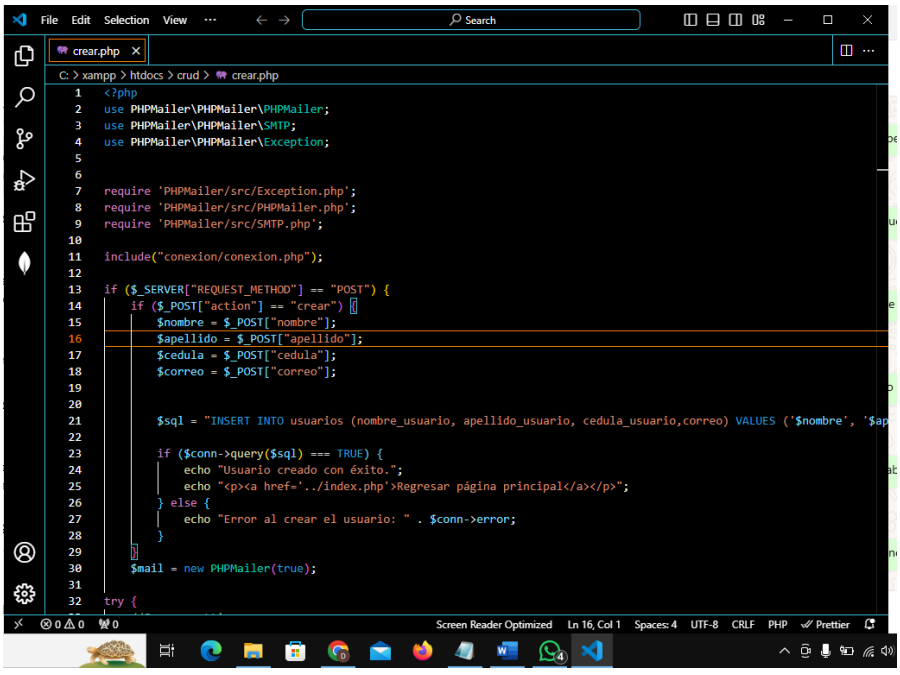

Figura 7. Código creado en PYCHARM

La imagen hace referencia al código utilizado para crear el CRUD en PHP.

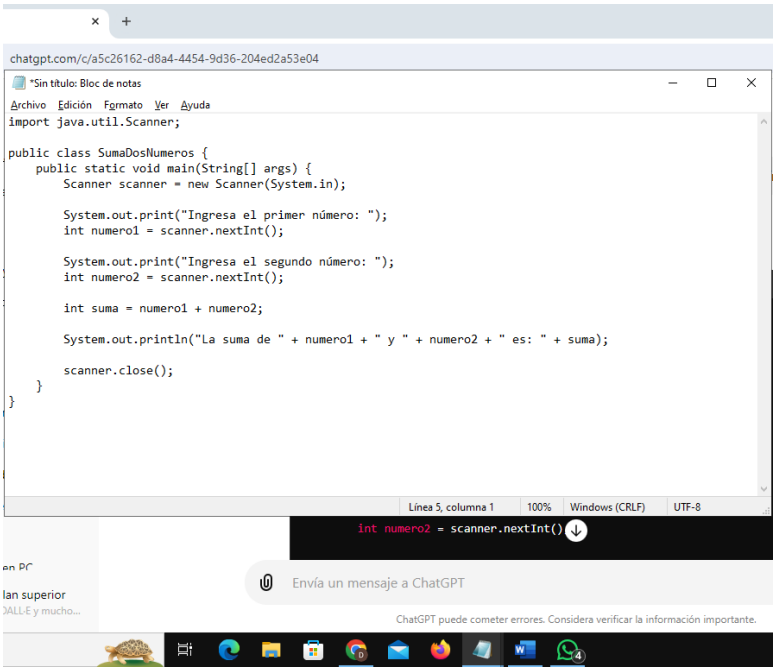

Figura 8. CRUD en PHP

Finalmente, la imagen referencia el código que se utilizó en NetBeans.

4.5 Análisis descriptivo

En la siguiente tabla se muestran los resultados que arrojaron los entornos de desarrollo con respecto a la accesibilidad, es importante aclarar que en este punto los usuarios no interactuaron con los entornos de desarrollo, el análisis comparativo con ellos se podrá observar en el capítulo siguiente, en este punto se muestra un análisis realizado desde la perspectiva del autor, comparando las últimas versiones de los entornos de desarrollo.

Tabla 2. Análisis descriptivo NetBeans

Requisito de accesibilidad	Resultado arrojado
Compatibilidad con lectores de pantalla	Mediante la creación de un programa que suma 2 números en NetBeans se evidenció que es compatible con los lectores de pantalla, las pruebas se realizaron mediante la lectura de cada una de las líneas de código arrojando un resultado exitoso
Navegación por teclado	NetBeans cuenta con muchos comandos que les permiten a las personas con discapacidad visual utilizar de forma sencilla el software. Cumpliendo así con este requisito
Lectura del resultado de la compilación	Al compilar el software se evidenció que NetBeans no es accesible para la interacción con el software creado, debido a ello para conocer la funcionalidad del software fue requerida la asistencia del apoyo visual.
Compatibilidad con tecnologías de asistencia	Al utilizar una línea Braille para realizar lectura del código los resultados fueron exitosos, NetBeans permitió el correcto manejo de este dispositivo cumpliendo satisfactoriamente el requisito de accesibilidad
Información semántica y estructurada	NetBeans cuenta con una estructura muy organizada de sus carpetas y configuraciones, esto permite que los usuarios en discapacidad visual puedan interactuar en los menús sin mayores complicaciones.
Documentación accesible	En el sitio web oficial de NetBeans se encontró una documentación con respecto al manejo del entorno de desarrollo mediante teclado, esto permite que los usuarios logren una interacción con el software exitosa en la mayoría de los aspectos.
Sonido en las líneas de código que contienen errores	Durante la calculadora se evitó que en NetBeans no se pueden identificar errores con señales sonoras, pero esto se puede mitigar con la lectura línea a línea antes de compilar algún archivo.

Fuente: autor

Tabla 3. Análisis descriptivo Visual Studio 2022

Requisito de accesibilidad	Resultado arrojado
Compatibilidad con lectores de pantalla	Las pruebas de compatibilidad se realizaron mediante la creación de un proyecto en .Net, los resultados fueron buenos, desde el proceso de instalación el software fue accesible y se mantuvo así hasta la finalización y compilación del código.
Navegación por teclado	Durante la realización del proyecto se lograron encontrar bastantes comandos para manejar el entorno de desarrollo mediante teclado, además cuenta con una gran documentación organizada

Requisito de accesibilidad	Resultado arrojado
	por temas para facilitarle al usuario la búsqueda
Compilación del código	El proceso de compilación no generó ningún problema de accesibilidad, esto hace que el software gane muchos puntos en el momento de ser utilizado
Información semántica y estructurada	Aunque la estructura de carpetas es accesible y organizada, no todo se encuentra fácilmente con el teclado, existen algunos archivos de configuración que solo pueden ubicarse con el mouse.
Documentación accesible	La documentación fue fácil de encontrar en el sitio web de Microsoft, esto facilitó mucho el aprendizaje de los comandos de teclado.
Compatibilidad con tecnologías de asistencia	Al utilizar la línea Braille no se generó ningún inconveniente al leer el código con el dispositivo.
Señal sonora en las líneas de código que tienen errores	Al realizar el proyecto no se encontró una configuración en el editor que permitiera detectar errores con sonido, además cuando existe alguno de estos el lector de pantalla no lo detecta complicando su corrección.

Fuente: autor

Tabla 4. Análisis descriptivo PyCharm

Requisito de accesibilidad	Resultado arrojado
Compatibilidad con lectores de pantalla	Las pruebas se realizaron con un proyecto creado, aunque la lectura del código fue accesible, no todas las funciones de este editor las detectó el lector de pantalla.
Navegación mediante teclado	Este editor cuenta con una alta gama de comandos para brindar accesibilidad, fue uno de los aspectos en los que más se destacó en la realización del proyecto
Compilación del código	En la realización de las pruebas se detectó que es muy Accesible para crear páginas web, sin embargo, al crear un programa más sencillo como una calculadora no es posible interactuar con las operaciones básicas con el lector de pantalla.
Compatibilidad con tecnologías de asistencia	Al utilizar el dispositivo Braille no generó ningún inconveniente para leer y manejar el entorno de desarrollo.
Documentación accesible	La documentación fue encontrada con facilidad, permitiendo así un rápido aprendizaje del manejo del IDE.
Información semántica y estructurada	La interfaz de usuario es muy accesible y bastante interactiva, fue fácil su manejo con el lector de pantalla

Requisito de accesibilidad	Resultado arrojado
Señal sonora en las líneas de código que contienen errores	No fue posible detectar los errores mediante algún sonido, además no es accesible la consola de comandos, sin duda fue uno de los puntos encontrar que tiene el editor de Python.

Fuente: autor

Tabla 5. Análisis descriptivo Visual Code

Requisito de accesibilidad	Resultado arrojado
Compatibilidad con lectores de pantalla	Mediante el desarrollo de una página web de hamburguesas con el entorno de desarrollo Visual Studio Code, se logró evidenciar una excelente compatibilidad con lectores de pantalla, ya que se logró leer por completo el código fuente de la aplicación, y también editarlo y modificarlo.
Navegación por teclado	Durante la creación de la página web, se pudo navegar por comandos de teclado, lo que permitió culminar el proceso.
Lectura del resultado de la compilación	El entorno de desarrollo respondió correctamente en el momento de compilar la página web, cuando se ingresó por consola el comando para compilar y construir la aplicación leyó en vos alta el paso a paso de la compilación.
Compatibilidad con tecnologías de asistencia	Al igual que los entornos de desarrollo anteriores, Visual Studio Code fue accesible con la línea Braille, permitiendo trabajar correctamente con este dispositivo.
Información semántica y estructurada	Los menús de configuración en Visual Studio Code se encuentran muy bien organizados, y es bastante sencillo acceder a ellos con el teclado, además las configuraciones están organizadas en pequeños menús permitiendo al usuario acceder a ellos fácilmente.
Documentación accesible	En la página web oficial del entorno de desarrollo se logró encontrar bastante documentación para configurar la accesibilidad, lo cual hace que cumpla este requisito.
Sonido en las líneas de código que contienen errores	En las configuraciones de accesibilidad de Visual Studio Code, fue posible encontrar las opciones para habilitar sonido en el momento de escribir código, facilitando al usuario corregir más fácil los errores presentados.

Fuente: autor

Conclusiones de la comparación entre entornos de desarrollo.

En términos generales los entornos de desarrollo cumplieron con muchos de los requisitos de accesibilidad establecidos, sin embargo, aún faltan requisitos por cumplir, pero

cabe resaltar que se evidencia el esfuerzo de los desarrolladores por querer mejorar sus páginas web y también los distintos (IDEs).

4.6 Resultados de las encuestas

En este capítulo se analizará la experiencia de usuario con cada uno de los entornos de desarrollo seleccionados, para determinar la escala de accesibilidad de estos softwares, se desarrollaron cuatro encuestas, una por cada entorno de desarrollo, los usuarios respondieron preguntas con respecto a su experiencia utilizando este tipo de programas, y también con base en los programas que se desarrollaron en los cuatro entornos de desarrollo. En el proceso de estudio de los entornos de desarrollo, cada usuario exploró tanto el (IDE) como el software que se desarrolló dentro de él, allí lograron explorar aspectos como:

Comandos por teclado, compilación de código, lectura por parte del lector de pantalla del código fuente y entre otros aspectos, los cuales permitieron determinar con mayor claridad qué tan accesibles son los distintos entornos de desarrollo para las personas con discapacidad visual. A continuación, se muestran los resultados de la encuesta analizando uno por uno los entornos de desarrollo.

Análisis de la población encuestada

El 60 % de las 10 personas que respondieron la encuesta fueron mujeres y el 40 % del género masculino, los rangos de edades estuvieron entre 18 y 30 años, pero el promedio más alto

se presentó entre 20 y 25 años, es decir, que la mayoría de los encuestados realizan carreras universitarias o afines.

Conocimiento en las áreas de programación por parte de los encuestados.

De las diez personas encuestadas hay que resaltar que ninguna se dedica profesionalmente a la programación, incluso algunas no tuvieron experiencia con ningún entorno de desarrollo, pero antes de la encuesta se les brindó el conocimiento necesario para diligenciarla, entre los aspectos enseñados a los encuestados destacan estos.

Comandos de teclado para utilizar los entornos de desarrollo, comandos de teclado para utilizar la consola, funciones esenciales de cada software entre otros aspectos importantes. Ningún usuario diligenció la encuesta sin conocer lo que se les preguntó, y antes de realizarla se evidenció que lograran utilizar adecuadamente el entorno de desarrollo, para que no obtengan respuestas incoherentes a las preguntas y alteren así el resultado del proyecto.

Ocupaciones de la población encuestada

En las entrevistas aplicadas a los usuarios para conocer su profesión, se logró determinar que cada uno se dedica a algo diferente. Dentro de las profesiones destacadas se encuentran: Psicología, comunicación social, música, derecho, ingeniería informática, lenguas extranjeras y contaduría. Cabe aclarar que las tres personas en el momento no tienen ocupación fija ni trabajo definido, como conclusión se destaca que, aunque las personas no conocían los entornos de desarrollo, lograron aprender rápidamente a utilizar sus funciones principales, porque manejan muy bien los lectores de pantalla cumpliendo así con el requisito previo para participar en la encuesta.

Análisis de los resultados obtenidos en NetBeans.

Diez usuarios con discapacidad probaron el entorno de desarrollo NetBeans, en dichas pruebas utilizaron los comandos de teclado del entorno de desarrollo y también utilizaron el software desarrollado en NetBeans para hacer el proceso de manera más dinámica.

En la primera pregunta de la encuesta se analizó el rango de edad de los usuarios y se obtuvieron los siguientes datos.

Figura 9. Rangos de edad de los participantes de la encuesta

En la Figura 9 se evidencia que el mayor rango de edad de los participantes de la encuesta está entre los 20 y 25 años, rango que corresponde a jóvenes dedicados a la programación.

En la segunda pregunta se indagó por el género de los usuarios, los resultados obtenidos fueron los siguientes.

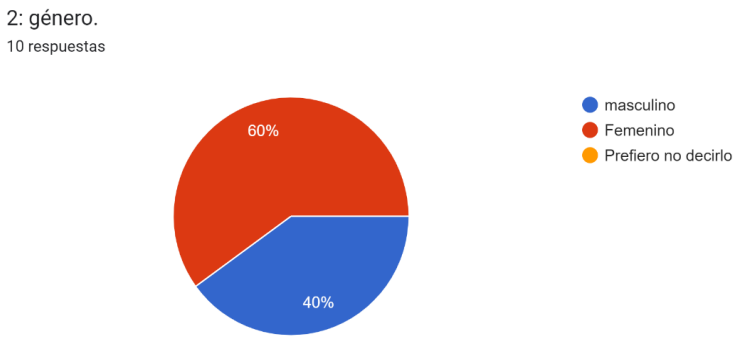

Figura 10. Género de los participantes de la encuesta NetBeans

En la siguiente pregunta se indagó por el lector de pantalla más utilizado por los usuarios, esto con el fin de determinar con cuál de ellos se sienten más cómodos para programar, y además también se busca comparar estos datos con los investigados en el capítulo 1, en el cual quedó determinado que JAWS es el lector de pantalla más utilizado, los datos obtenidos se observan en la siguiente gráfica.

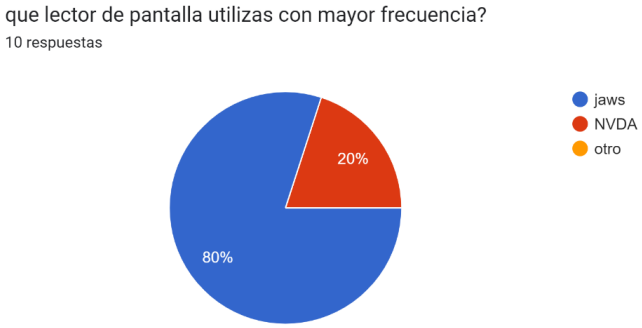

Figura 11. Lectores de pantalla usados por los participantes de la encuesta NetBeans

Los resultados fueron contundentes en esta pregunta, se determinó que 8 de 10 usuarios encuestados respondieron que utilizan el lector de pantalla JAWS, esto concuerda con los datos que indican que a nivel mundial el lector de pantalla más utilizado es el desarrollado por

Freedom Scientific, también se puede concluir que ningún usuario utiliza Linux ni Mac, debido a que su lector de pantalla no se encuentra marcado en ninguna respuesta.

Los resultados fueron los siguientes para determinar la experiencia de los usuarios con entornos de desarrollo, se les preguntó si antes habían utilizado alguno.

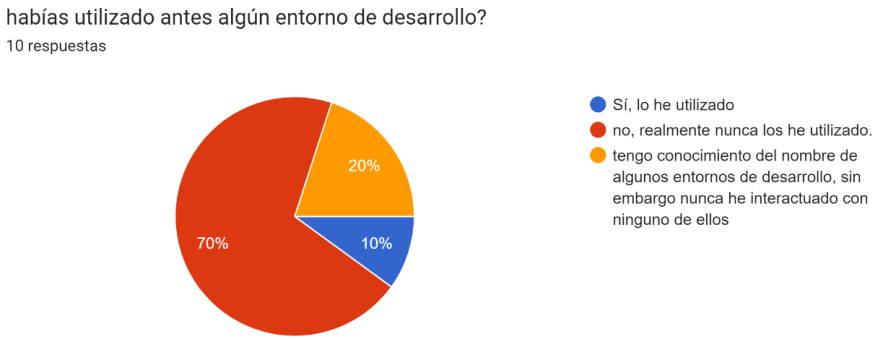

Figura 12. Uso de IDEs por los participantes de la encuesta NetBeans

En la gráfica se observa que la mayoría de los usuarios no conocen los entornos de desarrollo, es decir que su primera experiencia con ellos fue la obtenida en este estudio.

En la siguiente pregunta se desea conocer si los encuestados conocen el entorno de desarrollo NetBeans, los resultados fueron los siguientes.

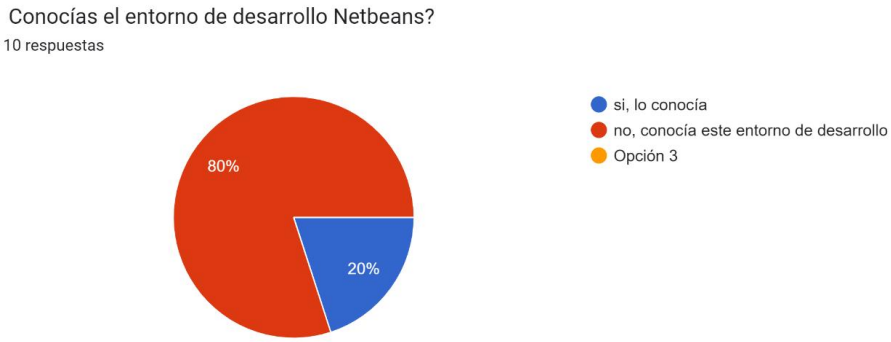

Figura 13. Conocimiento de NetBeans por los participantes de la encuesta

La gráfica deja ver que la mayoría de los usuarios no conocía este entorno de desarrollo.

En la siguiente pregunta se consulta por la accesibilidad del lector de pantalla JAWS con el entorno de desarrollo, los resultados fueron los siguientes.

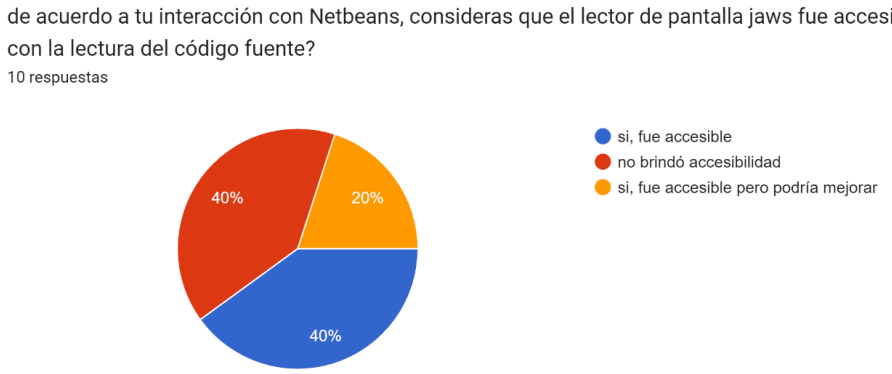

Figura 14. Uso de JAWS con NetBeans

En esta pregunta se puede observar división entre los encuestados, esto se debe a múltiples factores, recordemos que las experiencias de los usuarios con los entornos de desarrollo fueron personalizadas, por lo tanto, lo que para algunos puede parecer accesible para otros usuarios no lo es tanto, sin embargo, los resultados permiten concluir que faltan aspectos por mejorar dentro de Netbeans.

Los resultados fueron los siguientes, siguiendo los estándares establecidos por la W3C, se les preguntó a los usuarios por la organización de los menús en el entorno de desarrollo, o sea, si podían acceder fácilmente a los menús de configuración.

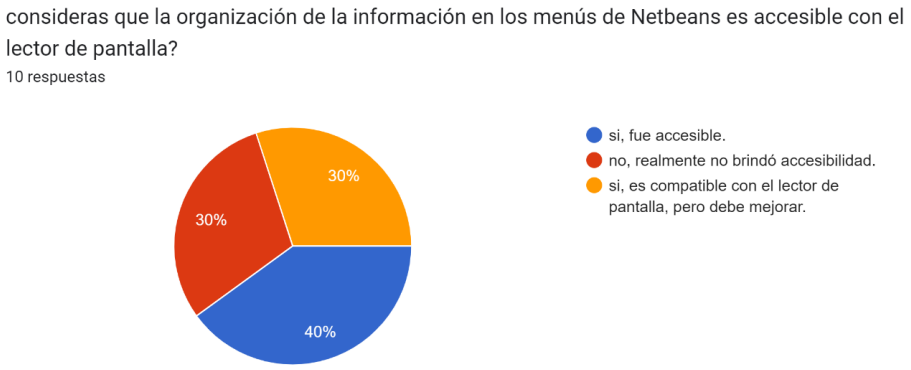

Figura 15. Menus en NetBeans

Al igual que la pregunta anterior, en esta también existió mucha división entre los encuestados, los factores que influyen son los mismos mencionados en la pregunta anterior.

En la siguiente pregunta se consultó a los usuarios por la accesibilidad en la compilación del código, los resultados obtenidos fueron los siguientes.

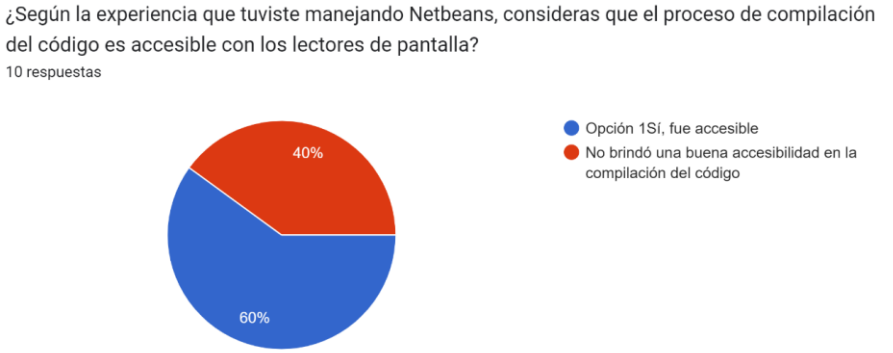

Figura 16. Compilación en NetBeans

En esta pregunta obtuvo una leve tendencia al sí, por un porcentaje muy pequeño los usuarios consideran que la compilación dentro del entorno de desarrollo si fue accesible, pero

algunos usuarios respondieron que la compilación del código no fue accesible, es decir, que hay aspectos buenos en el proceso, pero también factores que no permitieron satisfacer a los usuarios.

En la siguiente pregunta se buscó conocer qué tan fácil fue para los usuarios detectar errores en este entorno de desarrollo, los resultados obtenidos fueron los siguientes.

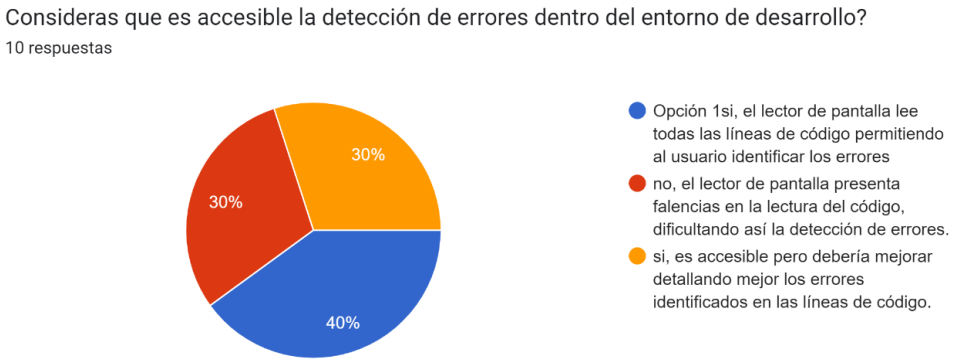

Figura 17. Detección de errores en NetBeans

De nuevo hay opiniones divididas en las respuestas, esto permite concluir que no es sencillo encontrar errores en el código manejando NetBeans, y deja saber que se debe mejorar bastante en este aspecto.

En la siguiente pregunta se indagó por la facilidad para encontrar comandos de teclado en el entorno de desarrollo, en la siguiente gráfica se muestran los resultados.

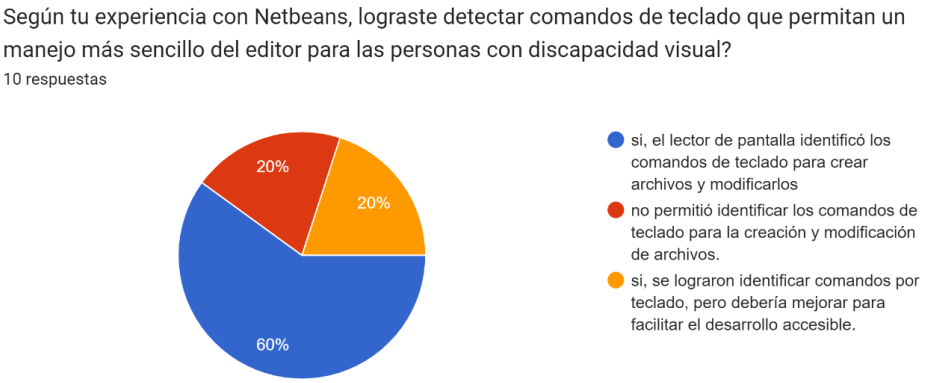

Figura 18. Comandos de teclado en NetBeans

En esta pregunta hay una leve tendencia al sí, los usuarios sí encontraron comandos para navegar por teclado, pero algunos de ellos no tienen conformidad en este aspecto.

Finalmente se consultó a los usuarios cuál aspecto faltaba por mejorar dentro del entorno de desarrollo, las respuestas fueron las siguientes.

Figura 19. Mejoras en NetBeans

Según los usuarios consultados el aspecto que más se debe mejorar es la estructuración dentro de los menús de NetBeans, algunos también recalcaron que falta mejorar la compilación del código.

Análisis de los resultados en Visual Studio Code.

Las preguntas que respondieron los usuarios con respecto a Visual Studio Code fueron muy similares a las de NetBeans, solo que en este caso lo único que se modifica es el entorno de desarrollo.

En la pregunta número 1 se pregunta a los usuarios por su edad, estos fueron los resultados obtenidos.

Figura 20. Rangos de edad de los participantes de la encuesta Visual Code

En la pregunta número 2 se consultó a los usuarios por su género, los resultados fueron los siguientes.

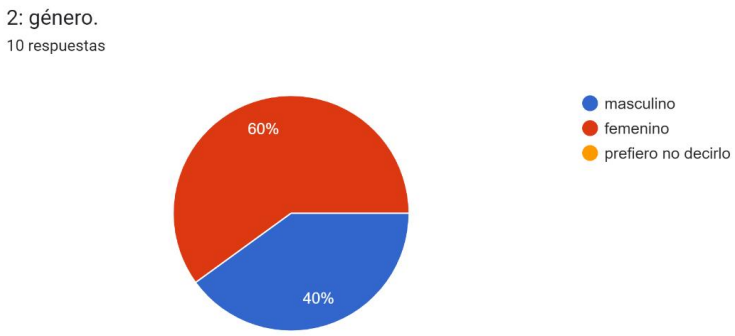

Figura 21. Género de los participantes de la encuesta Visual Code

En la pregunta número 3 se consulta por el lector de pantalla más utilizado para determinar con quién prefieren programar los usuarios, en el siguiente gráfico se muestran los resultados.

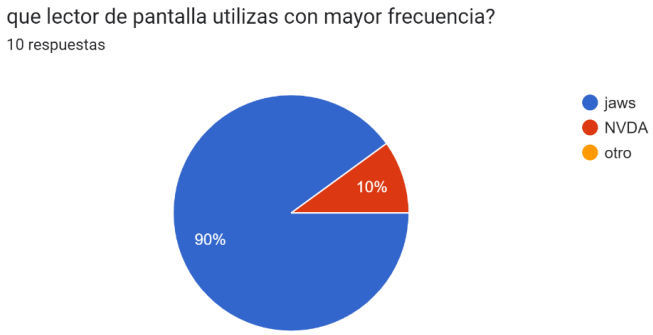

Figura 22. Lectores de pantalla usados por los participantes de la encuesta Visual Code

En la pregunta número 4 se pregunta a los usuarios si habían tenido contacto con algún entorno de desarrollo, los resultados obtenidos se muestran en la gráfica.

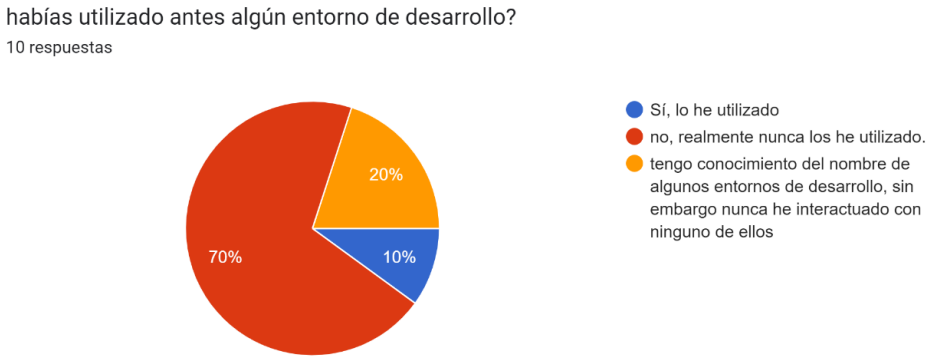

Figura 23. Uso de IDEs por los participantes de la encuesta Visual Code

Desde la siguiente pregunta en adelante se profundiza en Visual Studio Code, preguntando a los usuarios si lo conocían, estas fueron las respuestas.

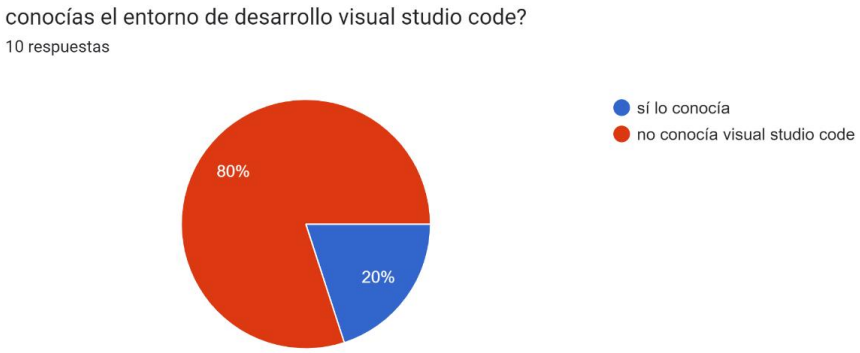

Figura 24. Conocimiento de Visual Code por los participantes de la encuesta

Al igual que en NetBeans, podemos evidenciar que los usuarios no tenían conocimiento de este entorno de desarrollo.

En la siguiente pregunta se indagó por la accesibilidad del lector de pantalla JAWS con el entorno de desarrollo, en el gráfico se observa que a los usuarios les pareció accesible, sin

embargo, la opción en la cual dice que hay aspectos a mejorar también tiene un número de personas que la marcaron, indicando de esta forma que aún existen aspectos que deben ser replanteados dentro del entorno de desarrollo para ofrecer una accesibilidad en su totalidad. A continuación, se muestra el gráfico con las respuestas.

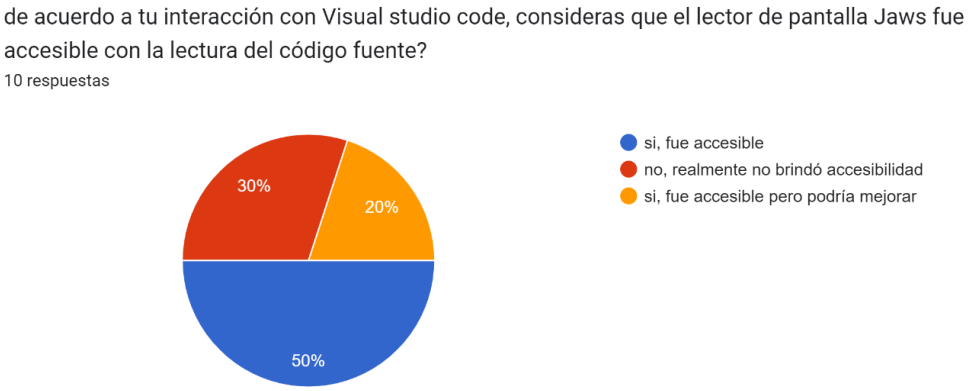

Figura 25. Uso de JAWS con Visual Code

En la siguiente pregunta se consulta a los usuarios por la organización de cada menú en la configuración del entorno de desarrollo, las respuestas obtenidas se observan en el siguiente gráfico.

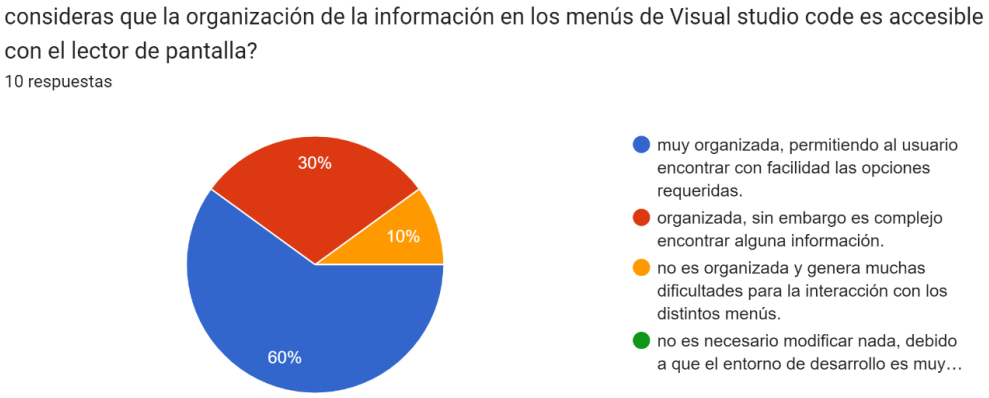

Figura 26. Menúes en Visual Code

En este aspecto Visual Studio Code obtuvo una gran aprobación por parte de los usuarios, ya que de 10 personas que respondieron la encuesta 6 marcaron que lo menús dentro del entorno de desarrollo eran muy organizados, y 3 marcaron que eran organizados, es decir que es relativamente sencillo encontrar configuraciones dentro de Visual Studio Code.

En la siguiente pregunta se les indagó a los usuarios sobre la accesibilidad en la compilación del código fuente, las respuestas fueron las siguientes.

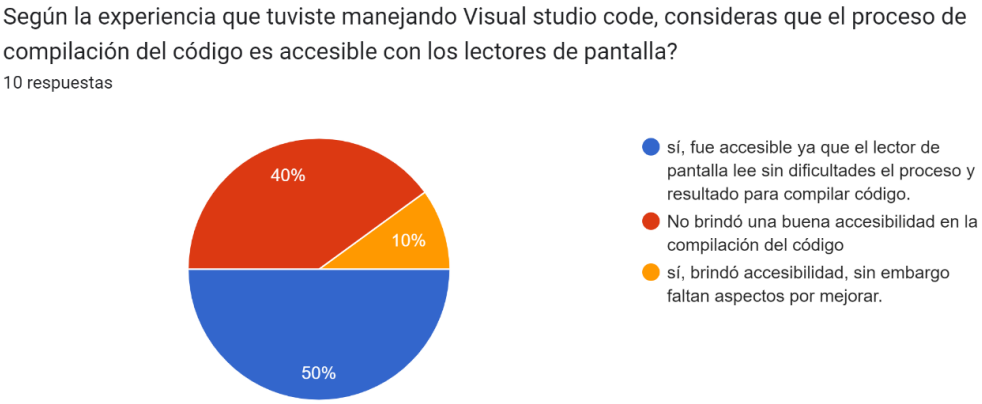

Figura 27. Compilación en Visual Code

Como se observa en la gráfica los resultados en esta pregunta estuvieron divididos, porque algunos usuarios tuvieron inconvenientes para leer los comandos de la consola con el lector de pantalla, esto generó que no les pareciera accesible el tema de compilar, pero algunos manifestaron que la consola de comandos les permitió realizar la compilación completa sin dificultad.

En la siguiente pregunta se buscó conocer la opinión de los usuarios con respecto a la detección de errores dentro del código, los resultados fueron los siguientes.

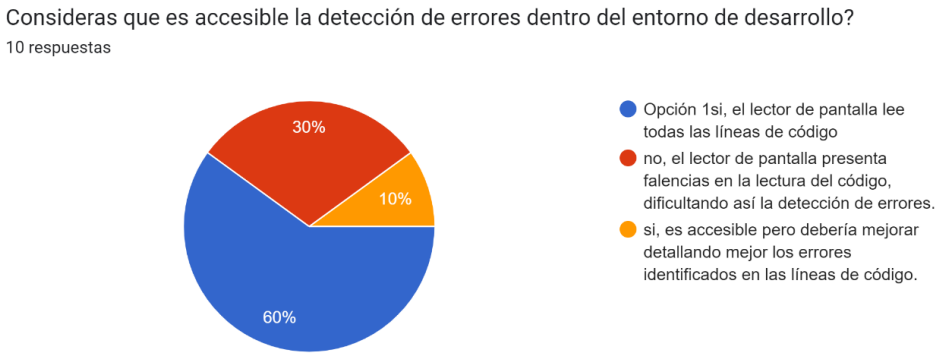

Figura 28. Detección de errores en Visual Code

En esta pregunta se puede evidenciar que para los usuarios sí fue accesible la detección de errores en el código, pero se debe considerar a los que dijeron que no lo era, aunque el número no es alto si se deben considerar algunas mejoras en este aspecto.

En la siguiente pregunta se consultó a los usuarios por la detección de comandos de teclado en el entorno de desarrollo, estas fueron sus respuestas

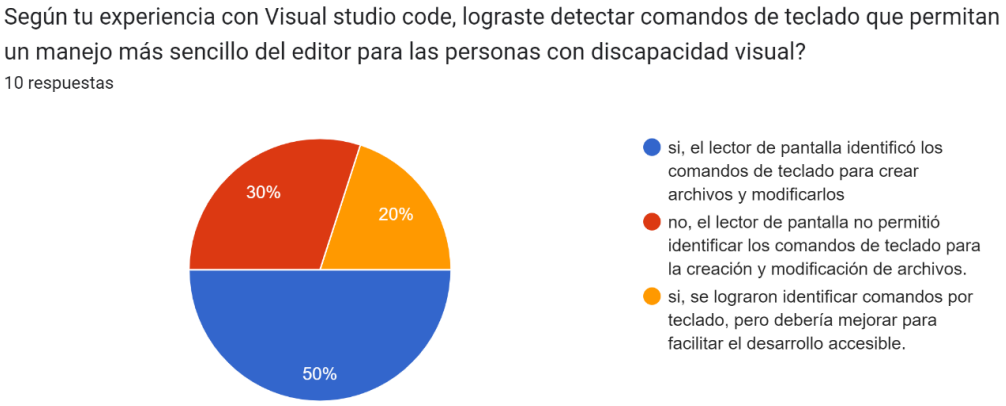

Figura 29. Comandos de teclado en Visual Code

En términos generales los usuarios detectaron comandos para utilizar Visual Studio Code de manera sencilla, 7 de 10 dijeron haberlos detectado y que si bien había aspectos a mejorar fue posible trabajar con los comandos.

Para finalizar se les preguntó a los usuarios cuál aspecto falta por mejorar dentro del entorno de desarrollo, las siguientes fueron sus respuestas.

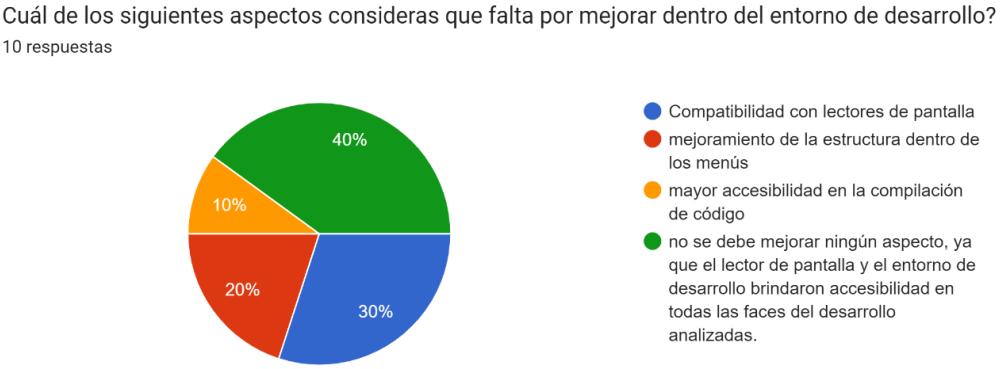

Figura 30. Mejoras en Visual Code

El gráfico observa que las opiniones están divididas, todas las opciones de mejora las seleccionaron al menos un usuario, pero 4 de ellos quedaron muy satisfechos con el entorno de desarrollo indicando que no se debe mejorar nada en él.

Análisis de los resultados en Visual Studio 2022.

Las primeras preguntas en la encuesta de Visual Studio 2022 son las mismas que en los 2 entornos de desarrollo anteriores, debido a ello no se enfatizará mucho en ellas.

En la primera pregunta se les consulta a los usuarios por su edad, estas fueron las respuestas.

Figura 31. Rangos de edad de los participantes de la encuesta Visual Studio 2022

En la siguiente pregunta se les consulta a los usuarios su género, estas fueron las respuestas

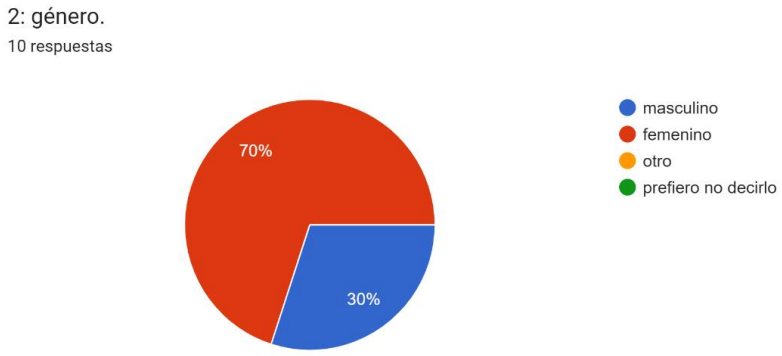

Figura 32. Género de los participantes de la encuesta Visual Studio 2022

Luego se consultó a los usuarios por el lector de pantalla que más utilizan, obteniendo las siguientes respuestas.

Figura 33. Lectores de pantalla usados por los participantes de la encuesta Visual Studio 2022

La próxima pregunta hacía referencia al conocimiento de los usuarios respecto a los entornos de desarrollo, en el siguiente gráfico se muestran las respuestas obtenidas.

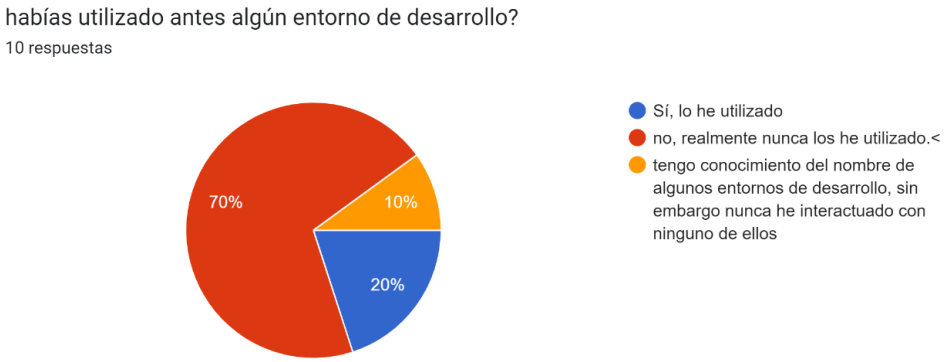

Figura 34. Uso de IDEs por los participantes de la encuesta Visual Studio 2022

A partir de la siguiente pregunta se enfocó en el entorno de desarrollo Visual Studio 2022, y para ello se les preguntó a los usuarios si lo conocían, las respuestas que se obtuvieron fueron las siguientes.

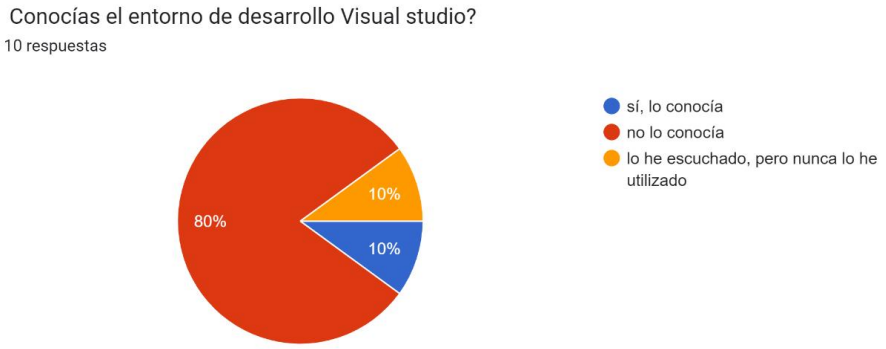

Figura 35. Conocimiento de Visual Studio 2022 por los participantes de la encuesta

Al igual que con los entornos de desarrollo anteriores los usuarios no tenían mucho conocimiento de él, su primera interacción fue respondiendo esta encuesta y explorando el entorno de desarrollo.

En la siguiente pregunta se consultó a los usuarios por la accesibilidad del código fuente con el lector de pantalla JAWS, estas fueron las respuestas obtenidas.

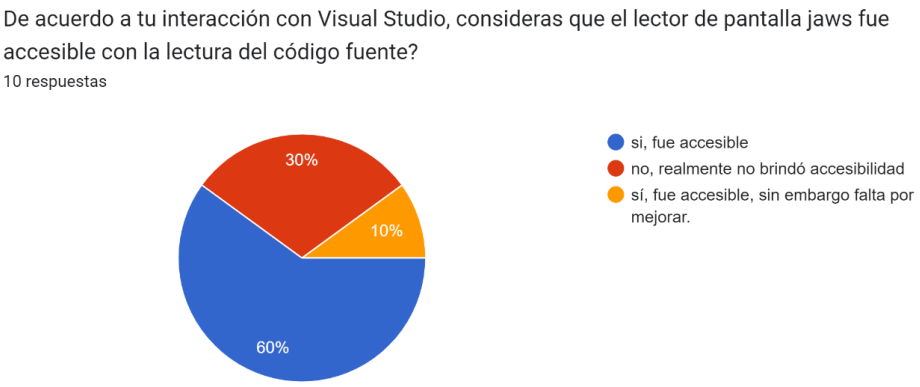

Figura 36. Uso de JAWS con Visual Studio 2022

En términos generales el entorno de desarrollo fue accesible para los usuarios en la lectura del código fuente, 6 de ellos dijeron que había sido fácil leer el código con el lector de pantalla, y uno afirmó que, si es accesible, aunque faltan aspectos a mejorar.

En la siguiente pregunta se indagó a los usuarios por la accesibilidad en el momento de navegar por el menú de opciones del editor, estas fueron las respuestas obtenidas.

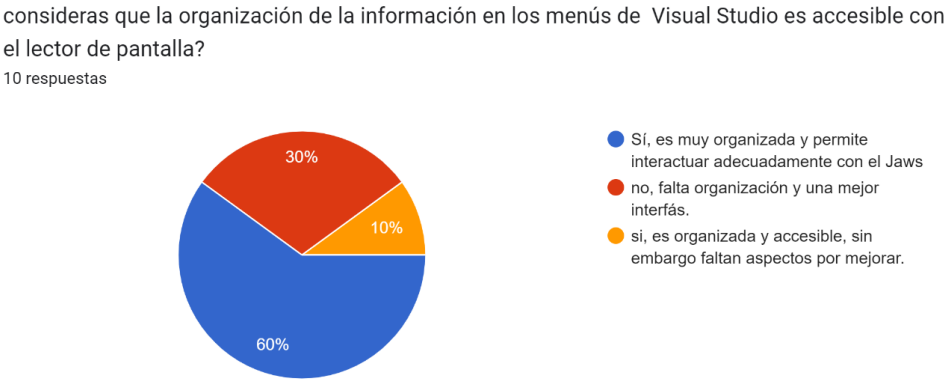

Figura 37. Menus en Visual Studio 2022

Al igual que en la pregunta anterior, la mayoría de los usuarios afirman que, si fue accesible en menú de Visual Studio, aunque el rango de inconformidad se mantuvo en un 10% con respecto a la pregunta anterior.

En la siguiente pregunta se consultó a los usuarios por la compilación del código utilizando el lector de pantalla, estas fueron las respuestas obtenidas.

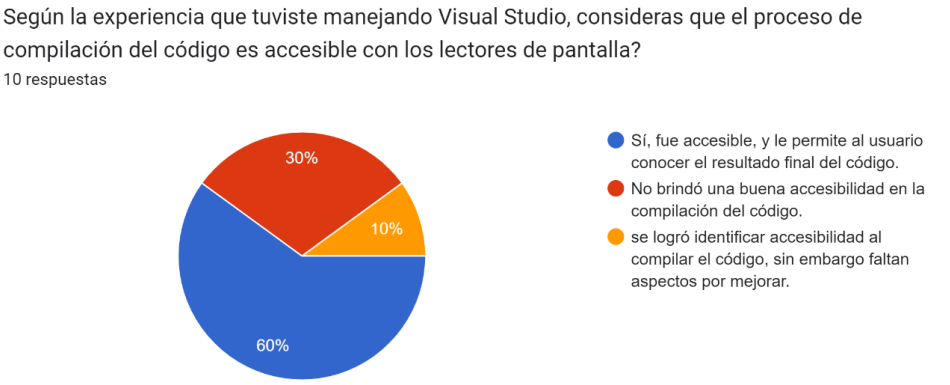

Figura 38. Compilación en Visual Studio 2022

El gráfico muestra que los usuarios lograron compilar el código con facilidad, aunque sigue la tendencia del mismo número de encuestados que no lograron tener satisfacción con este entorno de desarrollo.

En la siguiente pregunta se consultó por la detección de errores por parte de los usuarios al escribir código o leerlo, en el siguiente gráfico se reflejan los resultados.

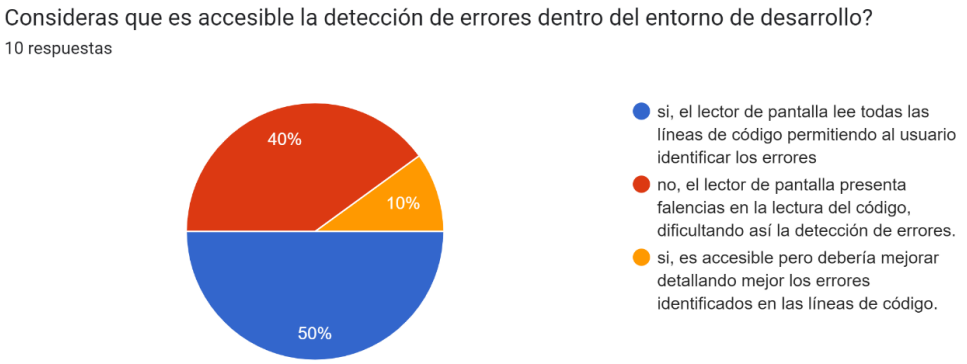

Figura 39. Detección de errores en Visual Studio 2022

En esta pregunta bajó un poco la aceptación por parte de los usuarios al entorno de desarrollo, esto se concluye porque la opción no cuenta con una respuesta más a su favor, sin embargo, la mitad de los encuestados logró detectar la forma de identificar errores en el código y corregirlos.

En la siguiente pregunta se consultó a los encuestados si para ellos fue sencillo detectar comandos de teclado dentro del editor, estas fueron las respuestas obtenidas.

Figura 40. Comandos de teclado en Visual Studio 2022

Continuando con la tendencia de este entorno de desarrollo, los usuarios en su mayoría afirmaron que Visual Studio permitió trabajar con comandos de teclado facilitando así su uso, el porcentaje de no aceptación se mantuvo en un 10%.

Finalmente, se les preguntó cuál aspecto faltaba por mejorar, obteniendo las siguientes respuestas.

Figura 41. Mejoras en Visual Studio 2022

La respuesta que prevaleció en esta pregunta fue la compatibilidad con lectores de pantalla, la mitad de los encuestados cree que falta por mejorar, pero hay un buen número que dijo que no se debía mejorar ningún aspecto, quedaron muy satisfechos con el entorno de desarrollo.

Análisis de resultados en PyCharm

Al igual que en los cuestionarios anteriores las primeras preguntas son iguales, por ende, no se profundizará en ellas.

En la primera pregunta se consultó a los usuarios por su edad, las respuestas fueron las siguientes.

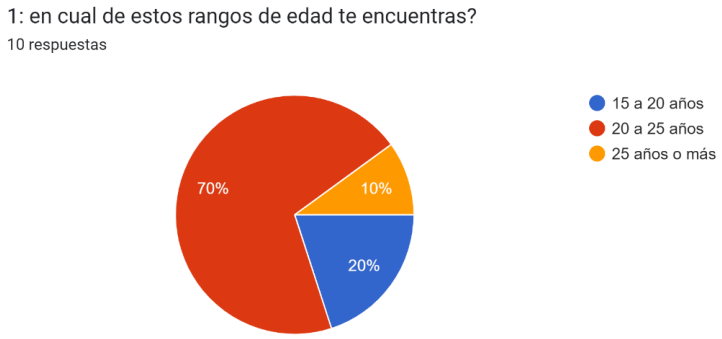

Figura 42. Rangos de edad de los participantes de la encuesta PyCharm

En la siguiente pregunta se consultó a los usuarios su género, las respuestas se observan en el siguiente gráfico.

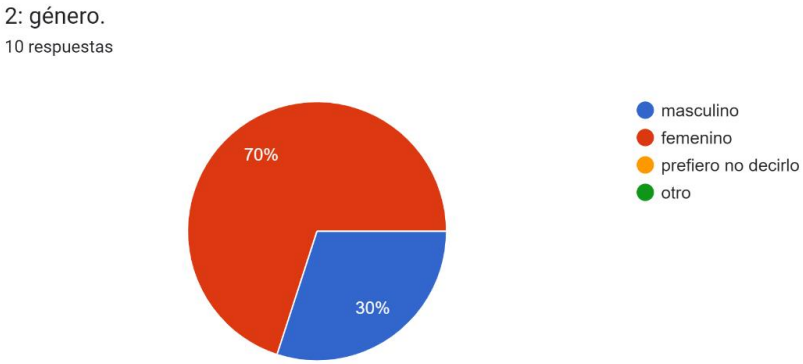

Figura 43. Género de los participantes de la encuesta PyCharm

En la pregunta número 3 se consulta a los encuestados por el lector de pantalla que más utilizan, las respuestas se observan en el siguiente gráfico.

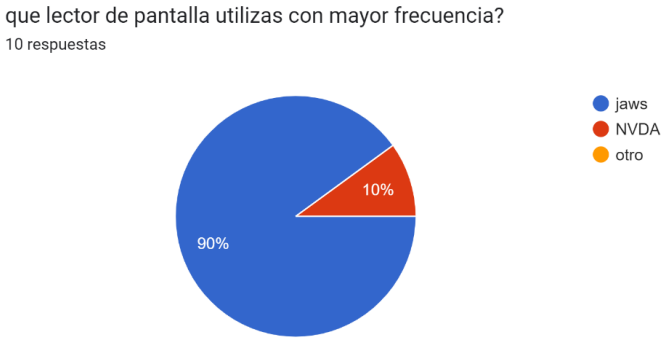

Figura 44. Lectores de pantalla usados por los participantes de la encuesta PyCharm

En la siguiente pregunta se consultó a los usuarios respecto a su conocimiento de los entornos de desarrollo, estas fueron las respuestas.

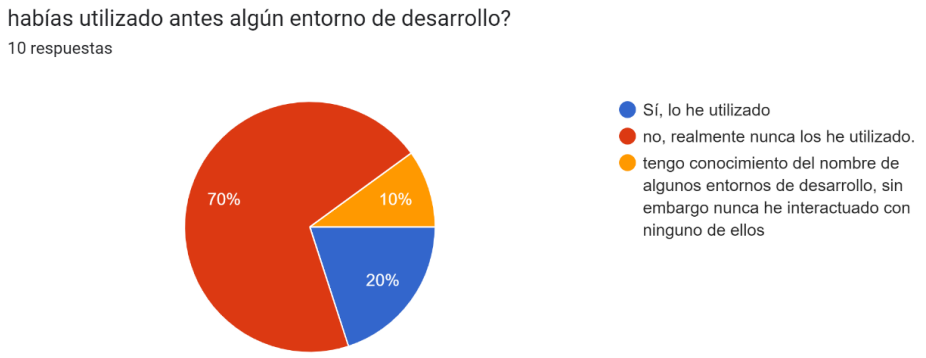

Figura 45. Uso de IDEs por los participantes de la encuesta PyCharm

A partir de esta pregunta, se enfocó en el entorno de desarrollo para trabajar en Python, y para ello se les preguntó a los usuarios si conocían el entorno de desarrollo, las respuestas se observan en el siguiente gráfico.

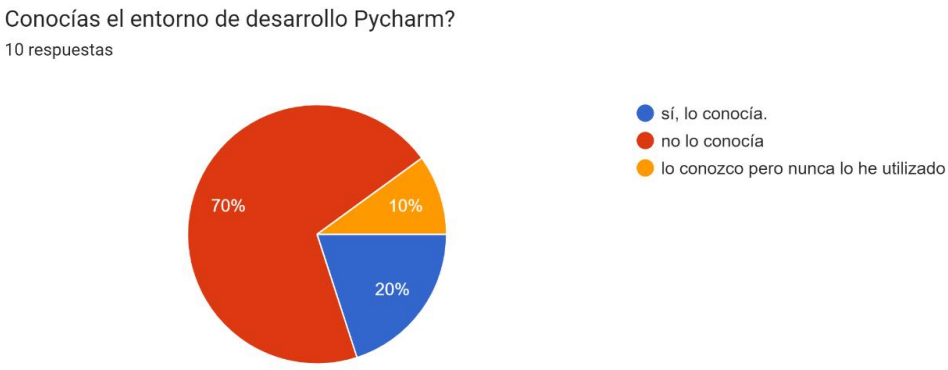

Figura 46. Conocimiento de PyCharm por los participantes de la encuesta

Igual que con los entornos de desarrollo anteriores, en esta pregunta se evidenció que los usuarios no conocían PyCharm, aunque algunos sí lo habían escuchado nombrar.

En la siguiente pregunta se indagó a los encuestados respecto a la accesibilidad del lector de pantalla JAWS con el código fuente, las respuestas se observan en el siguiente gráfico

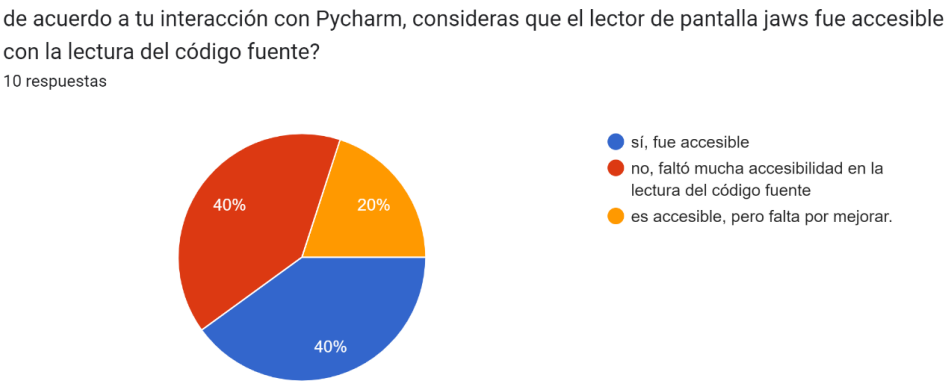

Figura 47. Uso de JAWS con PyCharm

En esta pregunta el entorno de desarrollo no cumplió con las expectativas de los usuarios, ya que no obtuvo la accesibilidad adecuada y no logró alcanzar la mitad de la aprobación de ellos.

En la siguiente pregunta se indagó a los usuarios por la organización de los menús de configuración, del entorno de desarrollo, en el siguiente gráfico se muestran las respuestas.

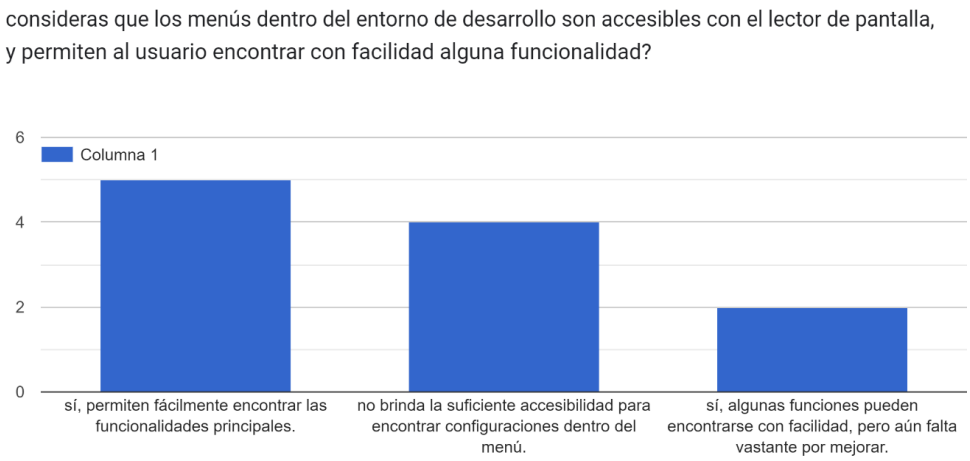

Figura 48. Menús en PyCharm

En términos generales los usuarios respondieron que sí fue posible acceder a los menús del entorno de desarrollo, sin embargo, la opción no también fue marcada, esto quiere decir que fue un poco complejo para las personas interactuar con la configuración de PyCharm.

En la siguiente pregunta se consultó a los usuarios por el proceso de compilar código, las respuestas se observan en el siguiente gráfico.

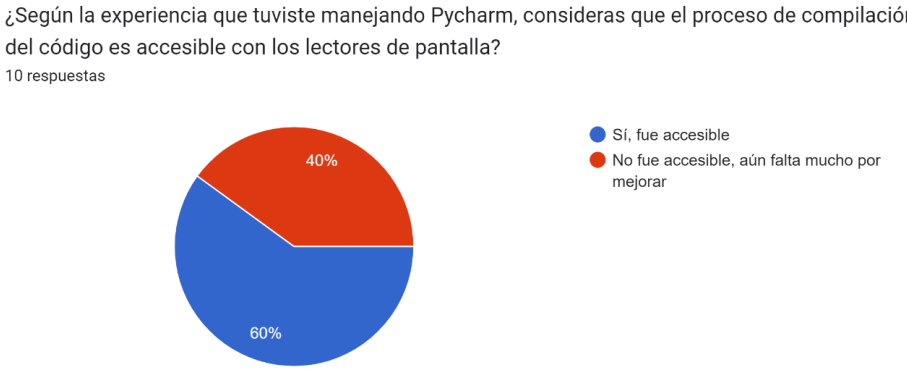

Figura 49. Compilación en PyCharm

En esta pregunta el entorno de desarrollo obtuvo aceptación por parte de los encuestados, aunque hay que resaltar que fue por un margen mínimo, lo cual permite concluir que interpretar código con Python no es del todo accesible.

En la siguiente pregunta se consultó a los usuarios si mediante la lectura del código con el lector de pantalla lograron detectar errores, o la manera de identificarlos con una señal sonora. Las respuestas se observan en este gráfico.

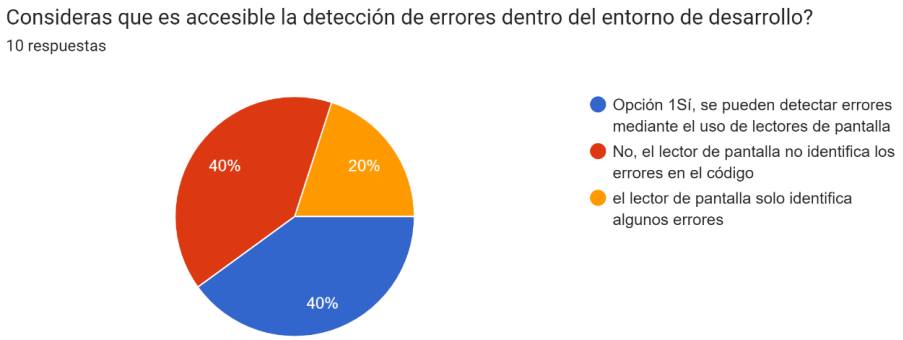

Figura 50. Detección de errores en PyCharm

Los usuarios respondieron, algunos contestaron que sí se podían identificar errores, otros afirmaron lo contrario, porque la experiencia es personal y cada uno mide la accesibilidad de forma distinta.

En la siguiente pregunta se consultó a los encuestados si lograron detectar comandos de teclado para trabajar con el editor, estas fueron las respuestas.

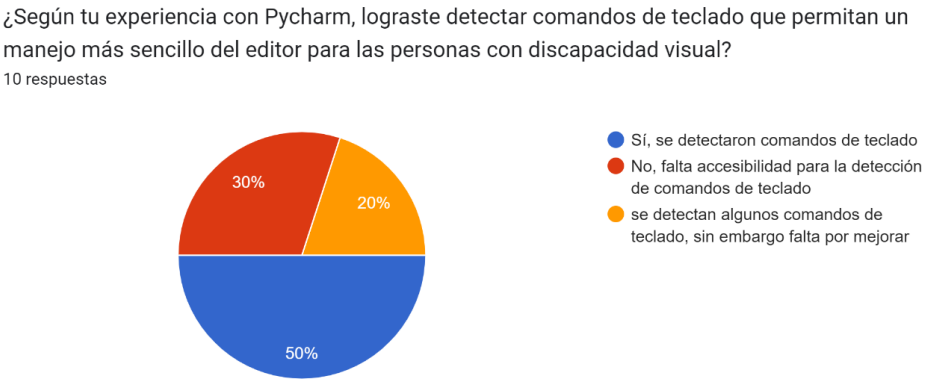

Figura 51. Comandos de teclado en PyCharm

Los usuarios identificaron comandos de teclado para trabajar con el editor, pero no todos quedaron satisfechos con los distintos tipos de comandos.

Por último, se preguntó a los usuarios cual aspecto se debería mejorar en este editor de código, obteniendo las siguientes respuestas.

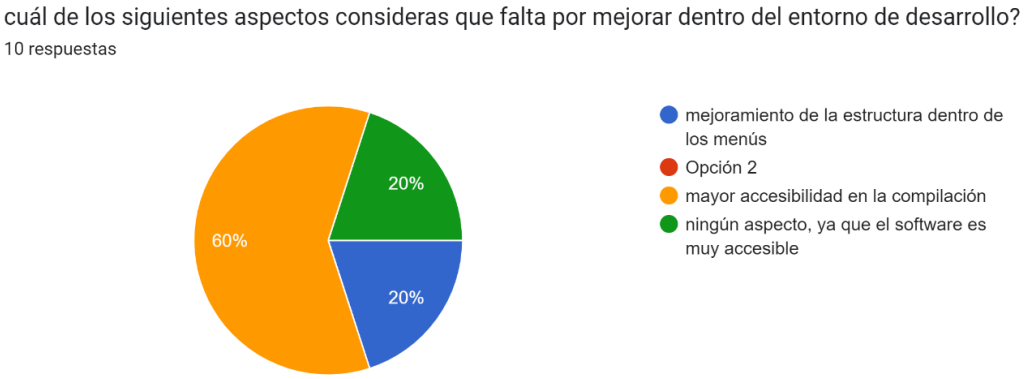

Figura 52. Mejoras en PyCharm

En esta pregunta los usuarios concluyeron que el mayor aspecto de este entorno de desarrollo en el que se debe trabajar es la compilación del código, esto se concluye debido a que más de la mitad de los usuarios respondieron esto.

Conclusiones del análisis

Ningún entorno de desarrollo obtuvo la completa aprobación de los usuarios, aunque se puede destacar que Visual Studio 2022 y Visual Studio Code obtuvieron los mejores resultados.

PyCharm fue el entorno de desarrollo con menor aprobación por parte de los encuestados, esto quiere decir que su accesibilidad no fue buena con el lector de pantalla.

El aspecto que más se destacó dentro de los entornos de desarrollo fue la facilidad para manejarlos por comandos de teclado, también algunos tuvieron buena accesibilidad para compilar el código.

5. Conclusiones

En este proyecto se analizan los aspectos a mejorar en cada entorno de desarrollo y se establecerán los lineamientos para mejorar la accesibilidad en cada entorno integrado.

Lo primero que se debe considerar antes de definir los lineamientos es que ya existe el estándar que deben seguir los entornos de desarrollo para ser accesibles, lo que se pretende con estos parámetros es reafirmar lo establecido por la W3C, y sugerir algunos aspectos a considerar a futuro para anexarlos a este estándar.

Lineamiento número 1. Los entornos de desarrollo deben permitir a los usuarios con discapacidad visual realizar la lectura del código fuente mediante un lector de pantalla, el texto debe poder ser editado, eliminado o simplemente leído utilizando las flechas del teclado.

Lineamiento número 2. Los entornos de desarrollo deben tener comandos de teclado para activar y desactivar la compatibilidad con los lectores de pantalla, estos deben estar disponibles para todo el público en sus páginas webs oficiales.

Lineamiento número 3. Si es necesario utilizar interfaz gráfica para realizar alguna acción dentro de un entorno de desarrollo, debe contar con un método que permita realizar la misma función con consola de comandos, y si el usuario con discapacidad desea realizar la acción mediante la interfaz gráfica, debe poder ejecutarse con el teclado.

Lineamiento número 4. Las consolas de comandos de todos los entornos de desarrollo deben permitir: Editar, leer, eliminar y crear comandos sin dificultad para el usuario con discapacidad visual, además, deben mostrar en pantalla el resultado del comando digitado para

que el lector de pantalla no lea aspectos innecesarios, como la hora de compilar, los archivos que se ejecutan. Entre otros aspectos, generan confusión en quien escucha el lector de pantalla.

Lineamiento número 5. Todos los botones en los entornos de desarrollo deben estar etiquetados, de lo contrario no serán accesibles a ningún lector de pantalla, debido a que lo leerá con la palabra (sin etiquetado) y no permitirá a la persona con discapacidad visual conocer cuál es su verdadera función.

Lineamiento número 6. Cuando se compile el código, el entorno de desarrollo deberá aclararse en el error o la solución que muestre en la pantalla, para evitar que el lector de pantalla lea información relevante al usuario.

Lineamiento número 7. En caso de ser posible los entornos de desarrollo deberían incluir un sonido en las líneas rojas que están marcadas con error dentro del código, los programadores sin discapacidad observan la línea marcada con color y rápidamente corrigen el error, la persona con discapacidad debe leer línea por línea el código intentando solucionarlo. Este aspecto genera retraso en la entrega del resultado final por parte del desarrollador ciego.

Lineamiento número 8. La persona con discapacidad visual debe poder desplazarse dentro de todos los menús de configuración del entorno de desarrollo, y además este debe permitir que por medio del teclado se realice cualquier tipo de configuración deseada por el usuario.

Lineamiento número 9. De la misma manera que existe un manual para utilizar el entorno de desarrollo para las personas sin discapacidad, así mismo debe existir un instructivo que les permita a las personas con discapacidad visual conocer la forma de utilizar su entorno de desarrollo preferido con diversos lectores de pantalla, al menos en el manual deberían estar incluidos JAWS y NVDA, debido a que son los que cuentan con mayor número de usuarios.

Nota: Estos parámetros recomendados para los creadores de entornos de desarrollo no son los únicos incluidos, ya que en todo el planeta hay muchos programadores ciegos que pueden proponer agregar muchos más, los escritos en este documento se realizaron considerando los comentarios de los usuarios que respondieron la encuesta del capítulo anterior.

El objetivo final de todo lo que se muestra en este documento es conocer opiniones de distintas personas con discapacidad visual que utilicen entornos de desarrollo y con base en ello, generar los lineamientos anexados anteriormente. Mediante ello se logrará concientizar a; lector final sobre la inclusión de los programadores ciegos en el mundo, generando así un mundo accesible para todos.

Conclusiones del objetivo 1

Después de la búsqueda de literatura respecto a la evolución de lectores de pantalla podemos concluir los siguientes aspectos.

Primero, en los inicios de los lectores de pantalla solo incluían lectura de texto plano sin considerar otro aspecto de los sitios web.

Después se incluyó la detección de imágenes y también mejoras en la lectura de caracteres especiales.

Y, tras incluir mejoras en las configuraciones de vos del lector de pantalla, se inició a trabajar con inteligencia artificial en cada uno, permitiendo así al usuario tener una excelente experiencia al utilizarlos.

La búsqueda literaria permite concluir que los lectores de pantalla continúan evolucionando continuamente, esto les permite a los usuarios desempeñarse en cualquier profesión que deseen. Con la inclusión del detector de imágenes en el lector de pantalla JAWS ya es posible obtener la descripción detallada de cada foto mostrada en la pantalla, permitiendo

la accesibilidad de los entornos para trabajar con diseño gráfico. Esto sin duda a facilitado la accesibilidad para todas las personas sin importar su profesión.

Conclusiones del objetivo 2:

Los entornos de desarrollo en sus inicios no incluyeron la accesibilidad con lectores de pantalla, en parte esto se debió a que los lectores de pantalla tampoco tenían la infraestructura necesaria para permitirlo, sin embargo, con el trascurso de los años y también de las actualizaciones de los entornos de desarrollo se logró hacerlos cada vez más accesibles para las personas ciegas.

Esto permite que en la actualidad más población con discapacidad visual pueda trabajar en las áreas de programación y de informática, sin embargo, aún falta por mejorar varios aspectos. Entre ellos destaca la accesibilidad en algunas consolas de comandos y también en algunas configuraciones en la edición de código. Pero aun así no se puede negar que las mejoras de accesibilidad en los entornos de desarrollo en los últimos años fueron notables, y lo más interesante es que sigue mejorando en conjunto con las actualizaciones de los lectores de pantalla, lo cual permitirá en un futuro una accesibilidad perfecta para todos.

Conclusiones del objetivo número3.

Después de realizar las encuestas con los usuarios respecto a la accesibilidad de entornos de desarrollo, y también de los lectores de pantalla podemos concluir lo siguiente.

Accesibilidad:

Dentro de la accesibilidad podemos destacar que los lectores de pantalla fueron accesibles con los entornos de desarrollo, la mayoría de los usuarios de las encuestas

manifestaron que el lector de pantalla logró detectar el código ilustrado en pantalla, además también fue posible utilizar comandos de teclado en la mayoría de los entornos de desarrollo.

Los entornos de desarrollo también incluyen alto contraste y legibilidad, permitiendo así a los usuarios leer con facilidad el texto. Respecto a la organización lógica y estructurada para permitirle a los usuarios navegar con facilidad por cada menú, se puede afirmar que en su gran mayoría los entornos de desarrollo cumplieron con este requisito.

En el aspecto que fallaron algunos entornos de desarrollo fue en el feedback de confirmación, algunos usuarios manifestaron que el lector de pantalla no les indicaba la activación de botones y tampoco la selección de algunos aspectos en el menú-.

A pesar de que este último punto no se cumplió en algunos entornos de desarrollo, al tabular los resultados de las encuestas realizadas a cada usuario se concluye que los resultados fueron exitosos, aunque se debe seguir trabajando para obtener mayor aprobación de cada persona en el momento de utilizar entornos de desarrollo, ya que, si bien pasaron la prueba de accesibilidad, no fue con el 100% de aprobación-

Conclusiones del objetivo número 4.

Requisitos que deben reunir los Lectores de pantalla:

Compatibilidad con múltiples plataformas: Debe funcionar en las plataformas principales como Windows, macOS y Linux para asegurar que los desarrolladores puedan elegir el sistema operativo que mejor se adapte a sus necesidades.

Soporte extensivo para aplicaciones: Debe ser capaz de interactuar de manera efectiva con una amplia gama de aplicaciones de desarrollo, incluidos IDEs, editores de código, herramientas de control de versiones, entre otros.

Interpretación precisa de elementos visuales: Debe interpretar con precisión y claridad todos los elementos visuales en la pantalla, incluidos textos, botones, menús, barras de herramientas, paneles de navegación, etc.

Navegación eficiente: Permitir la navegación rápida y eficiente por la interfaz de usuario mediante atajos de teclado y comandos específicos, facilitando así la productividad del desarrollador.

Personalización avanzada: Ofrecer opciones avanzadas de personalización para ajustar la velocidad de lectura, la voz utilizada, los tipos de información que se destacan (por ejemplo, errores de sintaxis, mensajes de depuración), entre otros.

Compatibilidad con tecnologías web y modernas: Ser compatible con tecnologías web modernas como AJAX, HTML5, CSS3, JavaScript, etc., para que los desarrolladores puedan trabajar con aplicaciones web complejas de manera efectiva.

Actualizaciones y soporte continuo: Debe ser actualizado regularmente para mantenerse al día con las últimas tecnologías y resolver problemas de accesibilidad que puedan surgir con nuevas versiones de software.

Requisitos que deben reunir los Entornos de Desarrollo Integrado (IDEs):

Compatibilidad con lectores de pantalla: Deben ser accesibles con lectores de pantalla comúnmente utilizados como JAWS, NVDA, VoiceOver, entre otros, asegurando que todos los elementos de la interfaz de usuario sean accesibles y manipulables.

Apoyo para atajos de teclado: Deben proporcionar una amplia gama de atajos configurables que permitan a los desarrolladores actuar rápidamente sin depender de la interacción manual con la interfaz gráfica.

Resaltado de sintaxis y errores: Deben ofrecer un resaltado claro y personalizable de sintaxis y errores, lo cual es crucial para la detección rápida de problemas y la escritura eficiente de código.

Documentación accesible: Deben incluir una documentación completa y accesible que explique todas las características del IDE, sus atajos de teclado, opciones de configuración y métodos recomendados para la navegación y manipulación eficientes.

Compatibilidad con plugins y extensiones: Deben admitir plugins y extensiones que amplíen las funcionalidades del IDE, asegurando que los desarrolladores ciegos tengan acceso a todas las herramientas necesarias para su trabajo.

Pruebas de accesibilidad: Deben ser sometidos a pruebas exhaustivas de accesibilidad con usuarios ciegos y con discapacidades visuales para identificar y corregir cualquier barrera de acceso que pueda existir.

Compromiso con la accesibilidad: Los desarrolladores de IDE deben mostrar un compromiso activo con la accesibilidad, integrando retroalimentaciones de usuarios con discapacidades visuales para mejorar continuamente la experiencia de usuario.

6. Referencias

AbilityNet, *(2024)*. AbilityNet, https://abilitynet.org.uk/news-blogs/voiceover-iphone-turns-10-and-turns-blind-access-11.

Accessibility Spark. (2024). How do screen readers work? https://accessibilityspark.com/how-do-screen-readers-work/

Accessing Higher Ground, (2022). *Survey of Screen-Readers inLinux Operating Systems,* https://accessinghigherground.org/survey-of-screen-readers-inlinux-operating-systems/.

Apori, I., & Apori, I, (2023). JAWS vs NVDA: Which Is Better? - Equally.AI Blog. *Equally.AI Blog*, https://blog.equally.ai/disability-guide/jaws-vs-nvda/.

Apple. Inc, (s. f.). Apple Developer Documentation, https://developer.apple.com/documentation/swiftui/accessibilitytechnologies/voiceover.

AudioEye.In, (s. f.). *Using the NVDA Screen Reader to Test Web Accessibility*, https://www.audioeye.com/post/nvda-screen-reader-web-accessibility/.

Blazie Engineering. (s.f.). Understanding screen readers. https://www.blazie.co.uk/understanding-screen-readers/

Bureau of Internet Accessibility, (2023). OrCa Screen Reader: An Overview for developers and content creators, https://www.boia.org/blog/orca-screen-reader-an-overview-for-developers-and-content-creators.

Carter, A, (2022). How to Use VoiceOver for Accessibility Testing, https://www.evoluted.net/blog/development/using-voiceover-for-accessibility-testing.

Comunidad NVDA, (s. f.). Accessibility Developer Guide, https://www.accessibility-developer-guide.com/setup/screen-readers/nvda/.

Comunidad NVDA, (s.f). NVDA 2024.2 Developer Guide, https://www.nvaccess.org/files/nvda/documentation/developerGuide.html.

DogQ. (2023). UI testing guide. https://dogq.io/blog/ui-testing-guide/#:~:text=UI%20(User%20Interface)%20testing%20involves,in%20response%20to%20user%20interactions

Eren, I, (2022). A brief introduction to JAWS, NVDA, and VoiceOver, https://css-tricks.com/comparing-jaws-nvda-and-voiceover/.

Gillis, A, (2022). *W3C (World Wide Web Consortium)*, https://www.techtarget.com/whatis/definition/W3C-World-Wide-Web-Consortium.

Han, S, (2022). The blind programmers who created screen readers. https://www.theverge.com/23203911/screen-readers-history-blind-henter-curran-teh-nvda.

JRebel By Perforce, (2020). *NetBeans Shortcuts and Cheat Sheet*, https://www.jrebel.com/blog/netbeans-shortcuts.

Knowbility. (2021). *A Brief History of Screen Readers*, https://knowbility.org/blog/2021/a-brief-history-of-screen-readers.

NV Access, (2024). NV Access, https://www.nvaccess.org/category/news/releases/.

Oracle Solaris, (2019). Orca Screen Reader Guide, https://docs.oracle.com/cd/E37838_01/html/E61045/orca-reader.html.

Orca Screen Reader and Magnifier, (2012). Oracle Solaris 11 Desktop Accessibility Guide, https://docs.oracle.com/cd/E26502_01/html/E29026/ats-2.html.

Orca Screen Reader. (s. f.). Orca Screen Reader, https://help.gnome.org/users/orca/stable/index.html.en.

Perkins School for the Blind, (2024). Perkins School For The Blind, https://www.perkins.org/resource/screenreader-comparisons/.

PyCharm, (s. f.). PyCharm Help, https://www.jetbrains.com/help/pycharm/mastering-keyboard-shortcuts.html.

Streetdirectory, (s.f). SG & Singapore Map, https://www.streetdirectory.com/travel/technology/computers/jaws-for-windows-a-short-history-lesson.php.

Supporting VoiceOver (s. f.). Apple Developer Documentation, https://developer.apple.com/documentation/uikit/accessibility_for_uikit/supporting_voiceover_in_your_app.

The American Foundation for the Blind, (s.f). *Screen readers*, https://www.afb.org/blindness-and-low-vision/using-technology/assistive-technology-products/screen-readers.

The GNOME Project, (s.f) Welcome to Orca, https://help.gnome.org/users/orca/stable/introduction.html.en.

The President and Fellows of Harvard College, (s. f.). Digital Accessibility, https://accessibility.huit.harvard.edu/nvda.

Ophthalmology24. (s.f.). Screen readers. https://www.ophthalmology24.com/screen-readers#:~:text=They%20use%20text%2Dto%2Dspeech,with%20and%20understand%20digital%20content

University of York, (s. f.). *IT Services*, https://www.york.ac.uk/it-services/tools/accessibility/text/.

Visual Studio Code. (2021). *Accessibility Visual Studio Code*, https://code.visualstudio.com/docs/editor/accessibility#:~:text=Make%20sure%20to%20have%20the,1%20as%20an%20environment%20variable.

Whitaker, E. (2019). *Advantages of Getting to Know JAWS*, https://blog.freedomscientific.com/advantages-of-getting-to-know-jaws/.

World Wide Web Consortium. W3C, (2023). *Web Content Accessibility Guidelines (WCAG) 2.1*, https://www.w3.org/TR/WCAG21/.

World Wide Web Consortium. W3C, (s. f.). *Web standards*, https://www.w3.org/standards/.