Informe de la Parte 01

Descripción de las Características Importantes de los Usuarios del Sistema

Los usuarios del sistema son individuos con discapacidades visuales que buscan aprender o mejorar sus habilidades en programación visual utilizando Python. Este grupo abarca desde principiantes hasta programadores experimentados que requieren herramientas accesibles debido a limitaciones visuales, como baja visión o ceguera total.

Visión General de los Objetivos de los Usuarios

Los usuarios buscan aprender y aplicar conceptos de programación visual utilizando Python. El sistema proporcionará una interfaz accesible que utiliza elementos auditivos, táctiles y visuales de alto contraste para facilitar el aprendizaje y la práctica de la programación. Esto es fundamental para brindar igualdad de oportunidades educativas y permitir que las personas con discapacidades visuales participen plenamente en campos técnicos.

Análisis de Tareas

1. Descripción de las Tareas y Especificaciones de Casos de Uso

Tarea Principal:

Aprender Programación Visual en Python

- **Usuarios**: Personas con discapacidades visuales.
- **Objetivo**: Facilitar el aprendizaje de conceptos de programación y lógica utilizando Python a través de métodos visuales adaptados, audio y táctiles.

Casos de Uso Específicos:

1. **Registro de Usuario**
    - **Actores:** Nuevo Usuario.
    - **Flujo Principal:**
        1. El usuario selecciona "Registrarse".
        2. Completa el formulario de registro.

3. Ajusta las configuraciones de accesibilidad iniciales.
4. El sistema valida y crea la cuenta.
5. El usuario recibe confirmación de la creación de la cuenta.

2. **Inicio de Sesión de Usuario**
   - **Actores:** Usuario existente.
   - **Flujo Principal:**
     1. El usuario introduce su nombre de usuario y contraseña.
     2. El sistema verifica las credenciales.
     3. El usuario accede a su cuenta.
3. **Modificación de Preferencias de Accesibilidad**
   - **Actores:** Usuario registrado.
   - **Flujo Principal:**
     1. El usuario accede a la configuración de su perfil.
     2. Modifica sus preferencias de accesibilidad.
     3. El sistema aplica y actualiza el perfil.
     4. El usuario recibe confirmación de la actualización.
4. **Creación de Código en el Editor Visual**
   - **Actores:** Usuario.
   - **Flujo Principal:**
     1. El usuario abre el Editor de Código Visual.
     2. Selecciona y arrastra bloques de código al área de trabajo.
     3. El sistema proporciona feedback táctil y auditivo.
     4. El usuario guarda su proyecto.
     5. El sistema confirma que el código ha sido guardado.
5. **Personalización del Entorno del Editor**
   - **Actores:** Usuario.
   - **Flujo Principal:**
     1. El usuario accede al menú de configuración del editor.

2. Ajusta el contraste, tamaño de fuente y opciones táctiles.
3. El sistema aplica los cambios.
4. El usuario recibe una notificación de los cambios aplicados.

6. **Acceso a Tutoriales Interactivos**
   - **Actores:** Usuario.
   - **Flujo Principal:**
     1. El usuario selecciona y accede a tutoriales interactivos.
     2. Sigue instrucciones paso a paso.
     3. Completa el tutorial.
     4. El sistema evalúa y proporciona feedback.
7. **Uso del Asistente de Voz y Chat**
   - **Actores:** Usuario.
   - **Flujo Principal:**
     1. El usuario activa el asistente de voz o chat.
     2. Formula preguntas o solicita ayuda.
     3. El sistema proporciona respuestas y asistencia.
     4. El usuario finaliza la interacción.
8. **Visualización del Análisis de Uso y Rendimiento**
   - **Actores:** Usuario.
   - **Flujo Principal:**
     1. El usuario accede al dashboard de análisis.
     2. El sistema muestra gráficos y métricas.
     3. El usuario interactúa con el dashboard para obtener detalles.
9. **Participación en Encuestas de Retroalimentación**
   - **Actores:** Usuario.
   - **Flujo Principal:**
     1. El usuario recibe una invitación para participar en una encuesta.
     2. Completa la encuesta.

3. El sistema recoge las respuestas.
4. El usuario recibe agradecimiento por participar.

Estos casos de uso proporcionan una visión clara de cómo los usuarios interactúan con las distintas funcionalidades de tu aplicación, facilitando el desarrollo y la implementación de cada módulo para garantizar que se satisfagan las necesidades de los usuarios de manera efectiva.

2. Descripción del Contexto de Desarrollo de Tareas

El contexto incluye el ambiente en el que los usuarios interactuarán con el asistente de estudios:

- **Entorno**: Hogar, escuelas, bibliotecas y otros lugares donde los usuarios podrían estudiar de manera independiente.
- **Dispositivos**: Computadoras personales, tabletas y dispositivos móviles adaptados con software y hardware de accesibilidad (pantallas táctiles, lectores de pantalla, auriculares).
- **Condiciones**: Variadas, desde entornos controlados y tranquilos hasta áreas más ruidosas, lo que puede influir en la preferencia por interacciones táctiles o auditivas.

3. Análisis de Tareas Estructurado del Problema

Descomposición de Tareas:

1. **Autenticación y Configuración de Perfiles**
   - Identificación y autenticación del usuario.
   - Personalización de la configuración de accesibilidad.
2. **Selección de Módulo y Navegación**
   - Exploración de cursos y módulos disponibles.
   - Selección de un módulo específico y comienzo de una lección.
3. **Interacción Educativa**
   - Participación en actividades de aprendizaje interactivo.
   - Uso del editor visual para programar, con soporte de accesibilidad integrado.
4. **Evaluación y Feedback**
   - Ejecución de programas escritos por el usuario.
   - Recepción de retroalimentación sobre errores y resultados.

5. **Apoyo Continuo**
    - Acceso a ayuda y tutoriales.
    - Opciones para feedback del usuario y soporte técnico.

Análisis de Sistemas Existentes

Vamos a analizar dos plataformas populares: Scratch y Code.org, enfocándonos en sus fortalezas y limitaciones, especialmente desde la perspectiva de accesibilidad para personas con discapacidades visuales.

1. Scratch

- **Descripción**: Plataforma de programación visual desarrollada por el MIT que permite a los usuarios crear programas utilizando bloques de código que se ensamblan como piezas de un rompecabezas.
- **Puntos Fuertes**:
    - Intuitiva y Fácil de Usar.
    - Comunidad Amplia.
    - Creatividad y Experimentación.
- **Limitaciones**:
    - Accesibilidad Limitada.
    - Limitaciones de Lenguaje.

2. Code.org

- **Descripción**: Plataforma educativa que ofrece una variedad de cursos y actividades de programación diseñadas para estudiantes de todas las edades.
- **Puntos Fuertes**:
    - Amplia Gama de Recursos.
    - Alta Disponibilidad y Gratuidad.
    - Enfoque Educativo Estructurado.
- **Limitaciones**:
    - Personalización Limitada.
    - Problemas de Accesibilidad.

Implicaciones para nuestro Proyecto

Identificamos áreas donde nuestro asistente puede mejorar la accesibilidad y la personalización, incluyendo soporte de voz y táctil, alto contraste y adaptabilidad visual, y la integración de tecnologías asistivas.

Criterios de Usabilidad Iniciales

Accesibilidad

- La aplicación debe ser utilizable por personas con diversos grados de discapacidad visual.
- Pruebas con usuarios para evaluar la usabilidad y el tiempo de

finalización de tareas.

Claridad

- La interfaz debe ser intuitiva y clara, minimizando la curva de aprendizaje.
- Observación de nuevos usuarios durante pruebas para medir la intuitividad de la interfaz.

Consistencia

- Los elementos de la interfaz deben ser coherentes en toda la aplicación.
- Revisión de la interfaz para asegurar consistencia y predecibilidad.

Eficiencia

- La aplicación debe permitir a los usuarios completar tareas de manera rápida y sencilla.
- Medición del tiempo que los usuarios tardan en completar tareas comunes.

Manejo de Errores

- La aplicación debe manejar errores de manera efectiva.
- Evaluación de la claridad y utilidad de los mensajes de error.

Flexibilidad

- La aplicación debe ofrecer múltiples formas de realizar tareas.
- Registro de las preferencias de los usuarios en la personalización de la experiencia.

Control del Usuario

- Los usuarios deben sentir que tienen control sobre la aplicación.
- Observación de la interacción de los usuarios para asegurar un sentido de control.

Soporte y Documentación

- Debe haber soporte y documentación accesibles y útiles.
- Evaluación de la accesibilidad y utilidad de la documentación.

Este enfoque asegura que la aplicación sea usable, accesible y efectiva para todos los usuarios, incluyendo aquellos con discapacidades visuales.

Recopilación de la Información

La información anterior fue compilada utilizando un enfoque basado en principios de usabilidad y diseño de interacción ampliamente reconocidos en la literatura de diseño de interfaces de usuario y experiencia de usuario (UX). No se utilizaron enlaces directos ni se citaron fuentes externas específicas. En cambio, la información se basa en estándares de la industria y prácticas recomendadas acumuladas a través de mi entrenamiento y conocimiento integrado en el campo del diseño UX y la accesibilidad.

Fundamentos del Diseño UX y Usabilidad

Los principios de usabilidad como claridad, eficiencia, manejo de errores, y flexibilidad son temas recurrentes en recursos autorizados en UX, como los propuestos por Jakob Nielsen y Don Norman, dos de los más influyentes teóricos en el campo de la usabilidad y el diseño de interacción. Estos principios se discuten en profundidad en obras como "Design of Everyday Things" por Don Norman y en varios artículos y directrices publicados por Nielsen Norman Group, un líder en investigación de experiencia de usuario.

Consideraciones de Accesibilidad

Las prácticas recomendadas en accesibilidad se basan en las directrices de la Web Accessibility Initiative (WAI) del World Wide Web Consortium (W3C), que proporciona un marco de trabajo para hacer los contenidos web accesibles para personas con discapacidades, incluyendo discapacidades visuales. Las recomendaciones del W3C, como parte de las Web Content Accessibility Guidelines (WCAG), sirven como referencia para diseñar interfaces accesibles.

Integración de Retroalimentación y Control de Usuarios

La importancia del control del usuario y la flexibilidad en el diseño de interfaces se basa en la necesidad de adaptar tecnologías para diversos usuarios, reflejando una perspectiva centrada

en el usuario que es fundamental para el diseño inclusivo. Esto se discute en contextos académicos y en publicaciones especializadas en diseño UX.

Validación y Pruebas de Usabilidad

La metodología para evaluar los principios de usabilidad mediante pruebas de usuario, observación y encuestas se basa en técnicas estándar de investigación en UX, que incluyen pruebas de usabilidad, evaluaciones heurísticas y estudios de campo. Estas técnicas son cruciales para obtener datos directos sobre cómo los usuarios interactúan con los sistemas y qué problemas pueden enfrentar.

Estos enfoques y métodos no sólo son esenciales para diseñar aplicaciones efectivas y accesibles, sino que también garantizan que los productos finales sean útiles, utilizables y accesibles para todos los usuarios, incluyendo aquellos con discapacidades.

Análisis Detallado de Usuarios

Usuarios

- **Descripción**: Usuarios con diversidad en el grado de discapacidad visual (baja visión a ceguera total).
- **Necesidades Específicas**:
  - Interfaces no visuales (auditivas, táctiles).
  - Retroalimentación accesible y comprensible.
  - Navegación simplificada y controles intuitivos.

Tareas

- **Aprender conceptos básicos y avanzados de programación** usando Python en un entorno visual adaptado.
- **Interactuar con un editor de código** que es compatible con tecnologías asistivas.
- **Comunicarse con asistentes o tutores virtuales** para obtener ayuda y retroalimentación.

Entorno

- **Entornos Diversos**: Desde hogares hasta centros educativos.
- **Requisitos de Tecnología**: Acceso a dispositivos compatibles con tecnologías asistivas (ordenadores, tablets).

Hipótesis Propuestas

1. **Hipótesis Principal**: Si el sistema ofrece interfaces multimodales (visual, auditivo, táctil), entonces los usuarios con cualquier grado de discapacidad visual podrán aprender programación de manera efectiva y autónoma.
2. **Hipótesis Secundaria**: La integración de retroalimentación instantánea y adaptativa mejora la comprensión de los conceptos de programación y reduce la frecuencia de errores en el código por parte de los usuarios.

Evaluación de Hipótesis

- **Análisis de los Datos de Usabilidad**: Observar cuánto tiempo y esfuerzo requieren los usuarios para completar tareas específicas y cuántos errores cometen.
- **Feedback de Usuarios**: Calificaciones y comentarios sobre la facilidad de uso y la eficacia del sistema en términos de aprendizaje de la programación.

Conclusiones e Inferencias

- **Confirmación de Hipótesis Principal**: Si los datos muestran que los usuarios pueden realizar todas las tareas de programación efectivamente utilizando diferentes modalidades, la hipótesis se confirma.
- **Reevaluación de Diseño**: Si los usuarios reportan dificultades con ciertos aspectos del sistema o si el tiempo para completar tareas es alto, estas áreas requerirán un rediseño.
- **Innovación Continua**: Las conclusiones pueden también llevar a la exploración de nuevas características o tecnologías que podrían incorporarse en futuras iteraciones del diseño.

Este análisis completo ayuda a asegurar que el diseño del asistente de estudios no solo es teóricamente sólido, sino que también es práctico y efectivo, basado en evidencia real y feedback directo de los usuarios a quienes está destinado a servir.