



Instituto Tecnológico de las Américas

Tarea III

Materia

Programación III | Grupo: sábado 9:00AM | Sección: 2

Tema

Módulo 5. Herramientas de administración de fuentes

Estudiante

Amelia Camila Branch Zorrilla | Matrícula: 2022-0008

Profesor

Kelyn Tejada Belliard

Fecha de entrega

15/12/2023

San Pedro de Macorís, República Dominicana

índice

Corrígeme	3
Cronograma del Proyecto	3
Primer reléase	5
Equipo Metodología Scrum	5
Acciones	7
Demostración.....	8
Plan de pruebas	9
Conclusiones	15
Bibliografía	16

Corrígeme

Utiliza el lenguaje de programación Python y LanguageTool, la herramienta gramatical de código abierto, también conocida como el corrector ortográfico de OpenOffice. Esta librería permite detectar errores gramaticales y ortográficos mediante un script de Python o a través de una interfaz de línea de comandos

Este proyecto de software está pensado en la práctica de la escritura para aquellos que estén aprendiendo español, las personas escribirán sus textos y el software se encargará de corregir sus faltas ortográficas y gramaticales junto con enseñarles que regla del español explica cada corrección que se hizo a su texto. Se planea tener listo el producto para su lanzamiento para finales de febrero del 2024.

Durante el proyecto se desarrollarán las siguientes características y funcionalidades:

- Reconocimiento de texto escrito
- Hacer sugerencias de correcciones
- Mostrar el texto original
- Presentar el texto corregido

Cronograma del Proyecto

	ACTIVITIES	EH	START	DUE	STATUS	%
	Fase 1: Planificación:	4h	27/Nov	29/Nov		100%
1	✓ Estudio de viabilidad	1h	27/Nov	27/Nov	Approved	100%
2	✓ Análisis de riesgos	1h	27/Nov	27/Nov	Approved	100%
3	✓ Planificación del proyecto con el equipo Scrum	2h	29/Nov	29/Nov	Finished	100%
	Fase 2: Análisis:	3.5h	28/Nov	29/Nov		100%
5	✓ Análisis de requerimientos del sistema	1.5h	28/Nov	29/Nov	Finished	100%
6	✓ Asignación de costes	2h	28/Nov	29/Nov	Finished	100%
	Fase 3: Diseño y estructura del software:	-	29/Nov	26/Jan		15%
8	⊙ Equipo de Desarrollo Scrum inicio y finalización ...	-	29/Nov	26/Jan	Started	15%
	Fase 4: Desarrollo de Software:	-	30/Nov	19/Feb		15%
10	⊙ Equipo de Desarrollo Scrum inicio y finalización ...	-	30/Nov	19/Feb	Started	15%
	Fase 5: Pruebas:	-	13/Dec	18/Feb		4%
12	⊙ Planificación de Pruebas	-	13/Dec	17/Feb	Started	4%
13	⊙ Pruebas al final de cada sprint	-	14/Dec	18/Feb	Started	4%

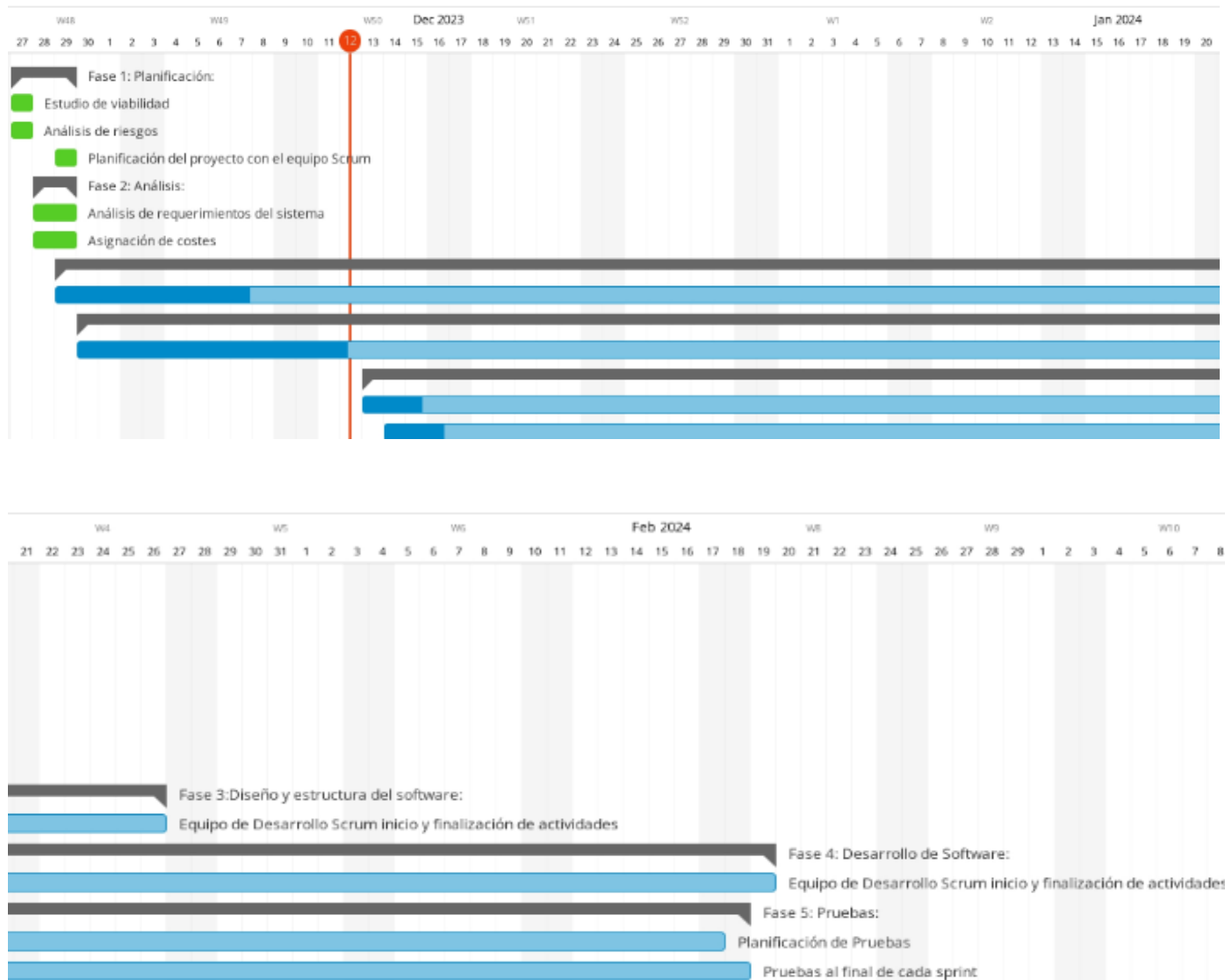


Diagrama de Gantt

(Cronograma del proyecto)

Primer reléase

En un primer lanzamiento el sistema va a poder analizar las faltas en textos de hasta más de 3 párrafos, podrá dar sugerencias y corregir el texto, sin embargo aun estaremos trabajando en la precisión de la corrección, tendremos que implementar nuevas tecnologías y librerías de Python como la NLTK, el kit de herramientas para procesamiento del lenguaje natural de Python para trabajar machine learning y entrenar el programa con más errores gramaticales y ortográficos que descubrimos que no reconoce la librería `lenguaje_tool_python`.

Requerimientos del sistema para el primer Release

- Reconocer un 70% de las faltas ortográficas y gramaticales
- Señalar los errores
- Mostrar el texto corregido
- Funcionar en Windows

Equipo Metodología Scrum

Tareas:

1. Desarrollar la aplicación de consola que permita introducir texto y corregir sus faltas ortográficas y gramaticales
2. Incluir la función de sugerencias de correcciones
3. Implementar una función que resalte o señalice los errores
4. Mostrar el texto original junto con el texto corregido después de cargar la corrección del texto introducido
5. Desarrollar la interfaz gráfica de la aplicación
6. Desarrollarla para funcionar en los sistemas operativos de Windows y Mac OS.
7. Desarrollarla para funcionar en Android y iOS

Equipo de trabajo

Gabriela Frías |Product Owner |Representante de negocio, gestiona el flujo de valor del producto a través del Product Backlog.

Habilidades: Flexible en su enfoque, piensa con rapidez, y es capaz de comunicar sus visiones a otras personas.

Luis Toribio|Scrum Master | Gestiona el proceso Scrum y asegura que el proceso Scrum se lleva a cabo correctamente, también ayuda a eliminar impedimentos que puedan afectar a la entrega del producto.

Habilidades: Dominio de la metodología Scrum, excelentes habilidades de comunicación, resolución de conflictos, conocimientos técnicos, gestión del tiempo y la planificación.

Equipo de Desarrollo

se encargará de crear un incremento terminado a partir de los elementos del Product Backlog seleccionados (Sprint Backlog) durante el Sprint Planning.

Alicia Abreau | Desarrolladora de software de escritorio,

Habilidades: Sistemas operativos, control de versiones, C#, Python, C++, Java bases de datos, experiencia de 8 años en el campo, trabajo en equipo, bilingüe

Christian Martínez | Desarrollador de inteligencia artificial/aprendizaje automático

Habilidades: R, Python, Java, control de versiones, bibliotecas de aprendizaje automático, experiencia de 4 años en el campo, trabajo en equipo, bilingüe, lingüista computacional.

Fernando Cabrera | Analista de calidad de software (QA)

Habilidades: Capacidad de trabajar con calma bajo presión, DevOps y metodología ágil, automatización, herramientas y técnicas de prueba, planificación y documentación de prueba, tecnología de la información y comunicación y programación, experiencia de 7 años en el campo.

Arianna Félix | Desarrolladora de Software

Habilidades: Programación Web, desarrollo de aplicaciones móviles, base de datos, Conocimiento de sistemas operativos, experiencia trabajando con metodologías ágiles, control de versiones, experiencia de 5 años en el campo.

Herramientas para usar

- Azure DevOps
- Scrum
- Python
- Paquete language_tool_python
- Paquete Tkinter

Épicas

Épica 1: Programa de corrección de textos

- Desarrollar una solución que corrija más del 70% de errores.

Épica 2: Interfaz gráfica

- Desarrollo de la interfaz gráfica.

Épica 3: Versión oficial

- Uso de la inteligencia artificial para mejorar la precisión en las correcciones, entrenando modelos.
- Desarrollo de diccionario para el programa
- Mejora de interfaz gráfica.

Acciones

Ceremonias Scrum

- mes 1:		4.25h	29/Nov	19/Dec		39%
1	✓ Sprint planning	2h	29/Nov	29/Nov	Finished	100%
2	✓ Daily meeting	0.25h	30/Nov	15/Dec	Started	40%
3	✓ Sprint review	1h	18/Dec	18/Dec	Not started	0%
4	✓ Sprint retrospective	1h	19/Dec	19/Dec	Not started	0%
+ Add task + Add section						
- mes 2:		4.75h	20/Dec	25/Jan		0%
7	✓ Sprint Planning	2h	20/Dec	20/Dec	Not started	0%
8	✓ Daily Scrum	0.25h	02/Jan	23/Jan	Not started	0%
9	✓ Sprint review	1.5h	24/Jan	24/Jan	Not started	0%
10	✓ Sprint Retrospective	1h	25/Jan	25/Jan	Not started	0%
+ Add task + Add section						
- Mes 3:		4.75h	26/Jan	21/Feb		0%
13	✓ Sprint planning	2h	26/Jan	26/Jan	Not started	0%
14	✓ Daily Scrum	0.25h	29/Jan	19/Feb	Not started	0%
15	✓ Sprint review	1.5h	20/Feb	20/Feb	Not started	0%
16	✓ Sprint retrospective	1h	21/Feb	21/Feb	Not started	0%
+ Add task + Add section						

Historias de usuario (Azure Devops)

Committed	1/5	Done
<div> <div>14 función de sugerencias de corrección</div> <div> Unassigned </div> <div>State <input checked="" type="radio"/> Committed</div> </div> <div>13</div>		<div> <div>15 Mostrar texto original</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>1</div>
		<div> <div>13 Función de corrección</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>13</div>
		<div> <div>16 Introducir cuadro de texto</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>3</div>
		<div> <div>22 Corrección de extranjerismos.</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>8</div>
		<div> <div>20 Corrección de mayúsculas al inicio de oración</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>13</div>
		<div> <div>19 Corrección de signos de puntuación.</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>8</div>
		<div> <div>17 Corrección de tildes</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>5</div>
		<div> <div>18 Diseñar interfaz gráfica de muestra</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>8</div>
		<div> <div>21 Corrección de verbos</div> <div> Unassigned </div> <div>State <input type="radio"/> Done</div> </div> <div>13</div>

Demostración

<https://youtu.be/g53bNJoPEys>

Video incremento

Plan de pruebas

Lista de Requerimientos Funcionales

1. Debe de mostrar sugerencias de corrección de errores.
2. Debe de mostrar el texto introducido como texto original.
3. Debe de identificar que después de punto va una letra mayúscula en un párrafo.
4. Debe de poder cambiar palabras como baseball, basketball, champagne a su escritura en español.
5. Debe de corregir los errores en cuanto a los signos de puntuación en los textos dados a corregir.
6. Debe de corregir la falta de tilde en cualquier tipo de palabra.
7. Debe de corregir verbos mal conjugados.

Lista de Requerimientos No Funcionales

8. Debe de corregir más del 70% errores encontrados en el texto.
9. Debe de funcionar a través de un cuadro de texto.
10. Debe de ser una interfaz gráfica clara y comprensible.

Criterios de aceptación de las pruebas

Si pasan más del 70% de las pruebas sin incidencias

Criterios de rechazo de las pruebas

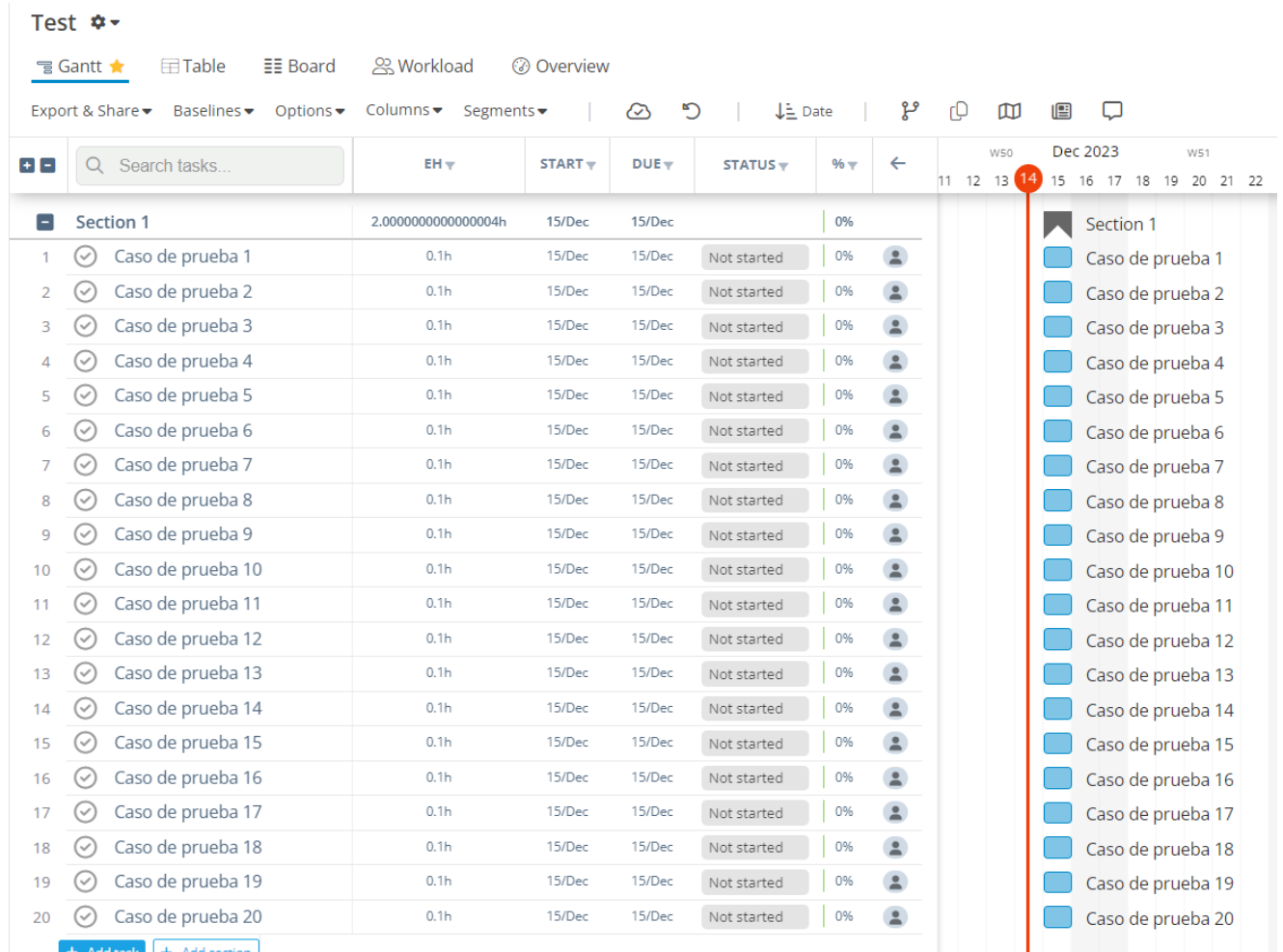
Si pasan menos del 70% de las pruebas

Herramientas

Unittest Python este programa aun no tiene su parte web por lo que no puedo usar Selenium, y teniendo el lenguaje de programación su propio entorno de pruebas prefería optar por este.

Cronograma de pruebas

El estimado es de 6min cada uno o menos.



Plantilla para cada caso de prueba

Plantilla de Casos de Pruebas de Software

Elaborado por: www.pmoinformatica.com

Proyecto: Corrígeme

Id	Caso de Prueba	Descripción	Fecha
1	Texto con errores ortográficos	Porcentaje de errores ortográficos corregidos	15/12/2023
2	Texto con errores gramaticales	Porcentaje de errores gramaticales corregidos	15/12/2023
3	Texto con errores O y G	Porcentaje de errores ortográficos y gramaticales corregidos	15/12/2023
4	Tildes	Se busca ver si el programa corrige cualquier error de tildes	15/12/2023
5	presentación de texto original	Se busca ver que el programa presente el texto original	15/12/2023
6	presentación de texto corregido	Se busca ver que el programa presente el texto corregido	15/12/2023
7	Sugerencias de corrección	El programa debe de presentar sugerencias de corrección	15/12/2023
8	Corrección de verbos mal conjugados	verificar que el programa corrija cualquier error de conjugación	15/12/2023
9	Mayúsculas	Se busca ver si el programa corrige cualquier error de uso de mayúsculas	15/12/2023
10	Extranjerismos	Se busca ver que el programa convierta palabras extranjeras a su escritura en español	15/12/2023
11	Signos de puntuación	Se busca ver si el programa corrige signos de puntuación	15/12/2023
12	Uso de la letra B y V	Se busca ver si el programa sabe diferenciar los usos de la b y v	15/12/2023
13	Uso de la letra Z, C, S	Se busca ver si el programa sabe diferenciar los usos de la Z, C, S	15/12/2023
14	Uso de la letra Y, LL	Se busca ver si el programa sabe diferenciar los usos de la Y, LL	15/12/2023
15	Uso de la letra Q, K, C	Se busca ver si el programa sabe diferenciar los usos de la Q, K, C	15/12/2023
16	Uso de la letra H	Se busca ver si el programa sabe diferenciar los usos de la H	15/12/2023
17	Uso de la letra G y J	Se busca ver si el programa sabe diferenciar los usos de la G y J	15/12/2023
18	Uso de la letra X y S	Se busca ver si el programa sabe diferenciar los usos de la X y S	15/12/2023
19	Uso de la letra Y y I	Se busca ver si el programa sabe diferenciar los usos de la Y y I	15/12/2023
20	Pruebas de usabilidad	Se busca ver si el programa tiene una interfaz de agrado para los usuarios.	15/12/2023

Plantilla con los equipos de pruebas y sus responsabilidades

Equipos de pruebas	Responsabilidades
QA y lingüista computacional	<ul style="list-style-type: none">Realizar las pruebas y verificar el correcto funcionamiento del programa.Tomar anotaciones de lo que funciona y lo que no.Revisar con el equipo lo que se puede mejorar.
Líder del equipo de desarrolladores y desarrolladores.	<ul style="list-style-type: none">El líder se encarga de supervisar las pruebas y de compartir los resultados con todo el equipo

Plan de automatización de pruebas

Objetivo de las Pruebas:

- Verificar que el programa de corrección de texto con interfaz gráfica funciona correctamente.
- Validar que las correcciones sugeridas se aplican correctamente al texto.

Herramientas de Automatización:

- Selenium (para interactuar con la interfaz gráfica).

Pasos de Prueba:

1. Prueba de Inicio de la Aplicación:

- **Objetivo:** Verificar que la aplicación se inicia correctamente.
- **Pasos:**
 1. Ejecutar el programa.
 2. Verificar que la ventana principal se abre sin errores.

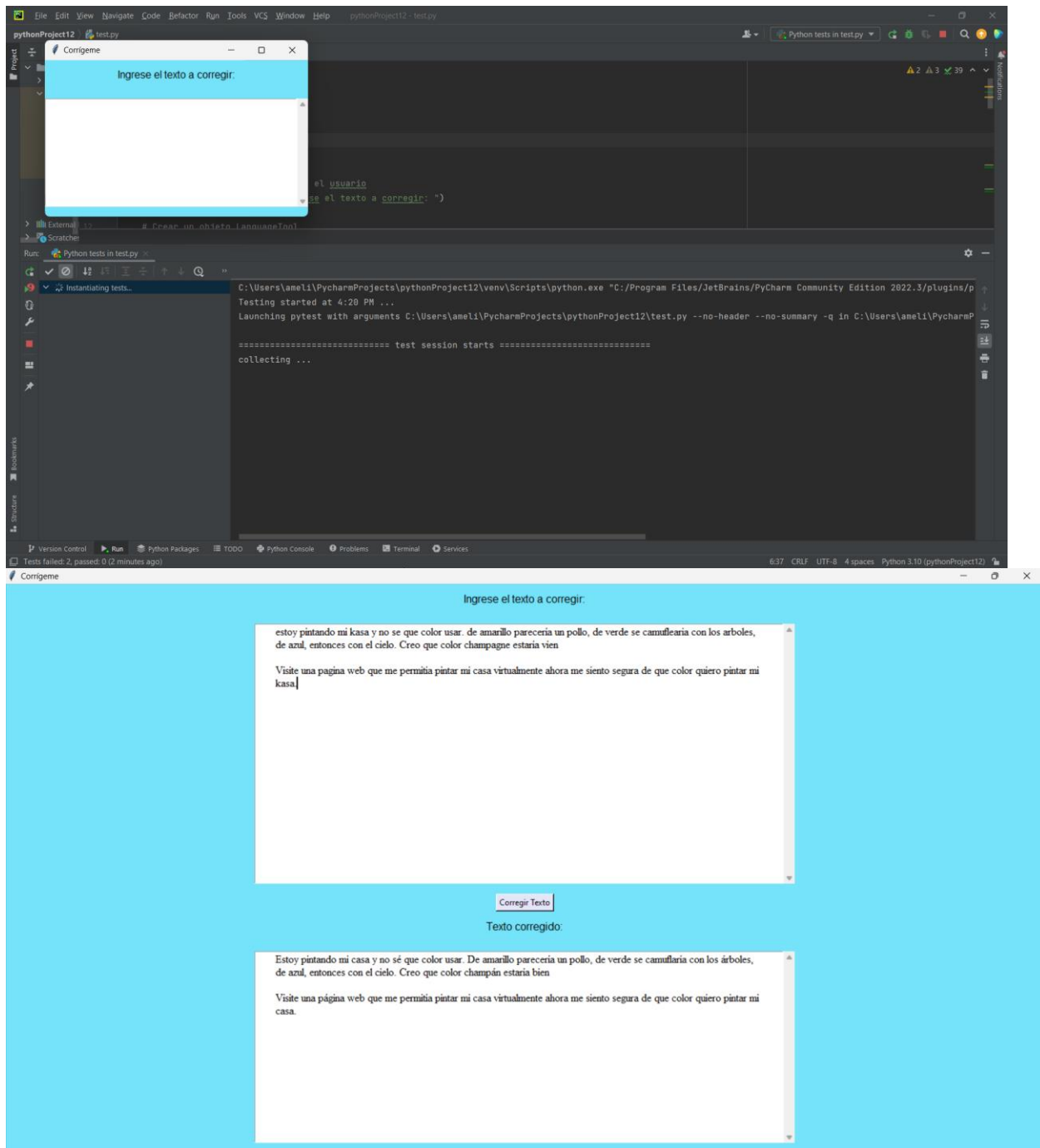
2. Prueba de Interacción con la Interfaz Gráfica:

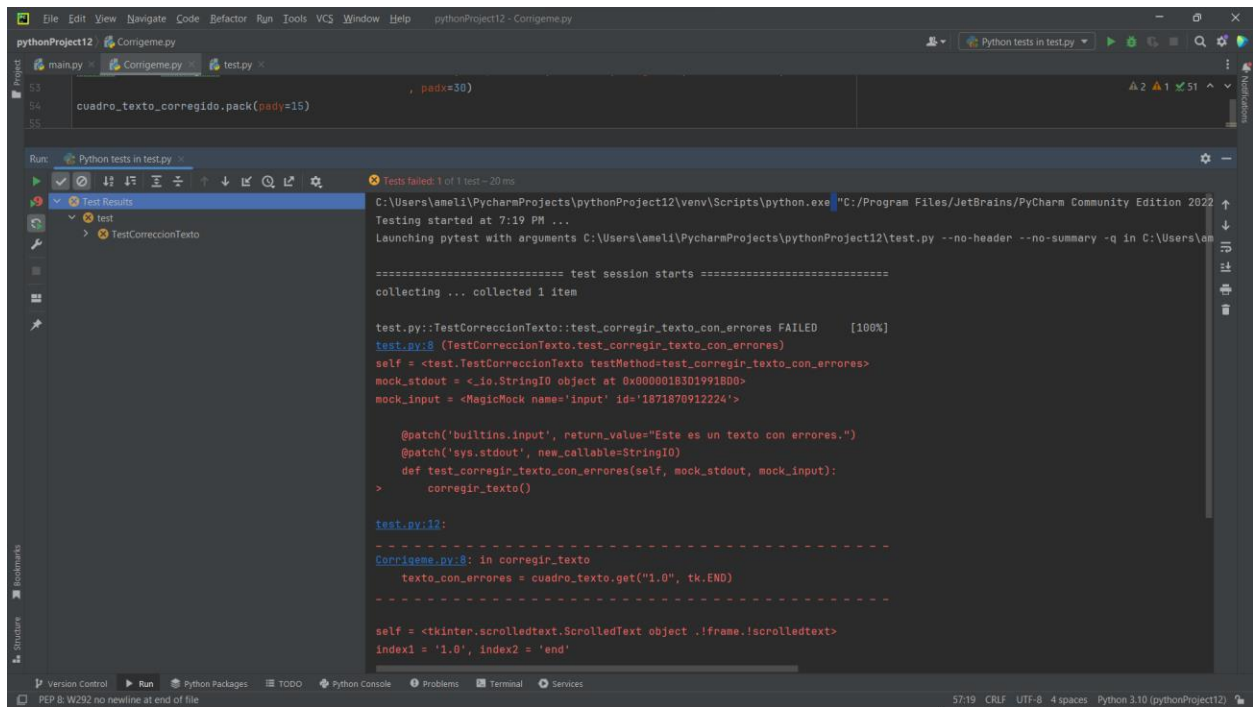
- **Objetivo:** Confirmar que la interfaz gráfica es interactiva y responde correctamente a las acciones del usuario.
- **Pasos:**
 1. Ingresar texto en el cuadro de texto de entrada.
 2. Hacer clic en el botón "Corregir Texto".
 3. Verificar que el cuadro de texto de salida muestra el texto corregido.

3. Prueba de Corrección de Texto:

- **Objetivo:** Asegurarse de que las correcciones sugeridas se aplican correctamente al texto.
- **Pasos:**
 1. Ingresar un texto con errores conocidos en el cuadro de texto de entrada.
 2. Hacer clic en el botón "Corregir Texto".
 3. Verificar que el cuadro de texto de salida muestra el texto corregido.
 4. Validar que las correcciones sugeridas se aplican correctamente.

Ejecutar y demostrar el plan de automatización de pruebas





Conclusiones

Este es un proyecto de gran dificultad debido a la necesidad del procesamiento del lenguaje natural a un programa computacional. Es difícil hacer que una máquina comprenda como hacemos los seres humanos todas las complejidades que conllevan analizar un texto. Sin embargo, con tiempo y dedicación se puede lograr que el proyecto corrija funcione en un 90% de correcciones de errores.

Actualmente no llega a un 70% sino a un 50% pero en un futuro podrá hacerlo.

Bibliografía

Canive, T. (2020, septiembre 24). *Épicas en Scrum: ¿qué es y cómo aplicarlas fácilmente?* Gestor de proyectos online. <https://www.sinnaps.com/blog-gestion-proyectos/scrum-epica>

Guzman, H. C. *Widget Text (Texto largo)*. Hektorprofe.net. Recuperado el 13 de diciembre de 2023, de <https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-text-texto-largo/>

McFarland, A. (2022, junio 14). *Las 10 mejores bibliotecas de Python para el procesamiento del lenguaje natural*. Unite.AI. Recuperado el 13 de diciembre de 2023, de <https://www.unite.ai/es/Las-10-mejores-bibliotecas-de-Python-para-el-procesamiento-del-lenguaje-natural/>

Overview. (s/f). CoreNLP. Recuperado el 15 de diciembre de 2023, de <https://stanfordnlp.github.io/CoreNLP/>

Pyspellchecker — pyspellchecker 0.7.2 documentation. Readthedocs.io. Recuperado el 14 de diciembre de 2023, de <https://pyspellchecker.readthedocs.io/en/latest/>

Tkinter Fonts. (s/f). Tutorialspoint.com. Recuperado el 14 de diciembre de 2023, de https://www.tutorialspoint.com/python/tk_fonts.htm

Vige, W. (2022, diciembre 3). *Puntos de historia: guía para estimar las historias de usuarios en Agile [2022]* • Asana. <https://asana.com/es/resources/story-points>