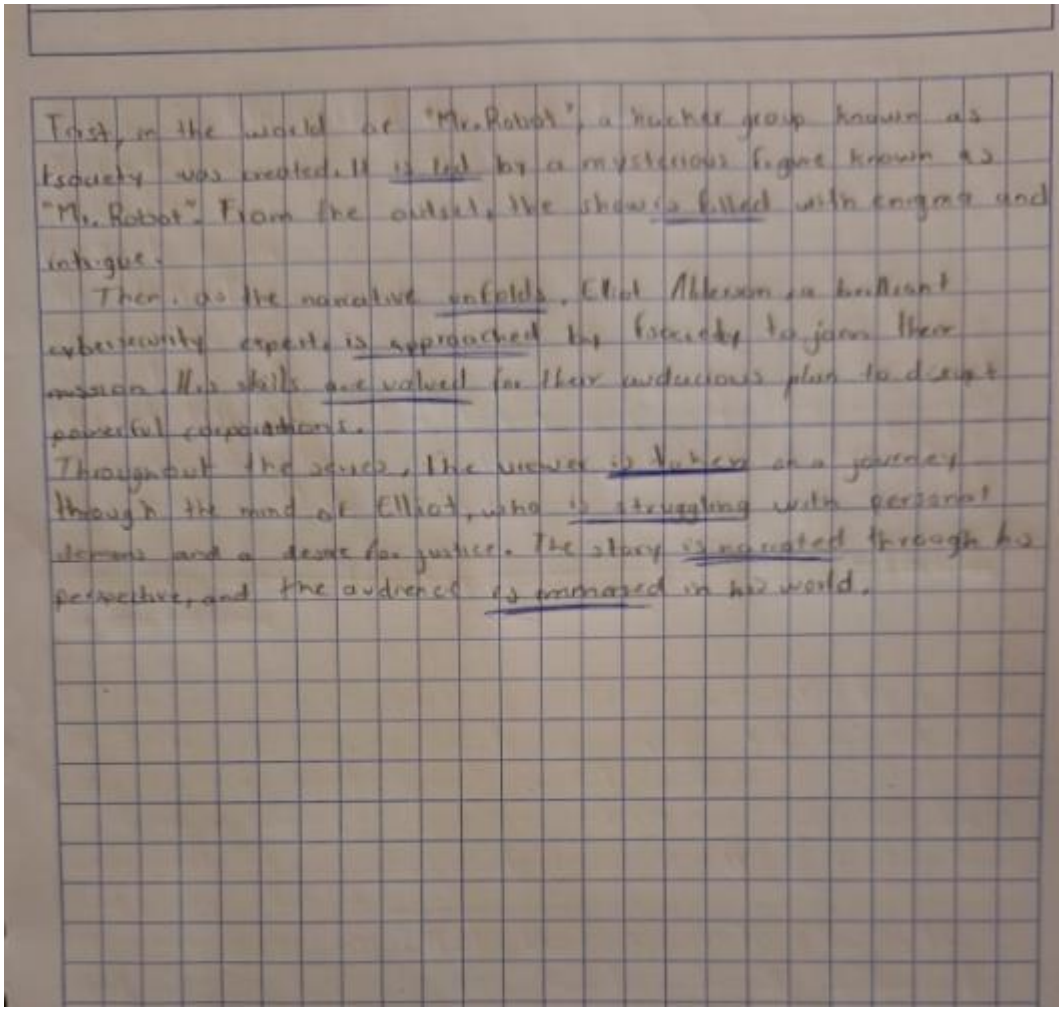
Chronological Connectors

First - Primero
Then - Entonces
Finally - Finalmente
After that - Despues de eso
After - Despues

First, the little boy went out to buy some candy. Then, he arrived to the store. After, he chose 3 different candies. After that, the kid went to pay. Finally, he went home



Writing

Introduction:

Significant changes have been observed in the programming process over the years. In this essay, the evolution of programming will be explored, emphasizing the use of passive voice in the present and past to highlight the transformations that have occurred.

Body:

Before:

First, in the early days of programming, code was primarily written in low-level programming languages. Extensive understanding of the hardware architecture was required, and each line of code had to be manually entered into the computer. Errors were often introduced and required meticulous debugging. Then, the debugging process was laborious, with code errors identified and rectified. Finally, in the event of program updates or changes, making modifications to the existing code was a daunting task, often leading to unintended consequences.

Now:

In contrast, programming has undergone a remarkable transformation. First, the advent of high-level programming languages and integrated development environments has simplified the process. Then, the use of version control systems and collaborative tools now allow multiple programmers to work on the same project simultaneously, significantly improving code quality and efficiency. After that, automated testing and debugging tools have reduced the time and effort required to identify and correct errors in the code. After, the concept of modular and object-

oriented programming has enabled developers to reuse code, making it easier to update and maintain software. As a result, the programming process has become more efficient and less error-prone.

Conclusion:

In conclusion, the evolution of programming from its complex and manual origins to its current state of automation and efficiency is evident. These changes underscore the continuous improvement and innovation within the field of programming.

Speaking:
Person A: Hey, have you ever thought about how much programming has changed over the years?

Person B: Absolutely! It's fascinating to see the evolution. In the early days, code had to be manually entered into computers in low-level languages. Errors were common and debugging was a painstaking process.

Person A: True, and making changes to existing code was quite a challenge. Now, though, things have drastically changed. High-level languages and integrated development environments have simplified the process.

Person B: Yeah, and with version control systems and collaborative tools, multiple programmers can work on the same project simultaneously. It significantly improves code quality and efficiency.

Person A: After that, automated testing and debugging tools have reduced the time needed to identify and correct errors. It's made the whole process more streamlined.

Person B: And let's not forget about modular and object-oriented programming. Reusing code has become so much easier, making it a breeze to update and maintain software.

Person A: The transformation is remarkable. Programming today is characterized by efficiency and collaboration, creating higher-quality software in less time.

Person B: Absolutely. These changes underscore the continuous improvement and innovation within the field of programming.