**Bibliografía y Recursos Educativos**

He creado un programa completo de Airport Traffic Manager con todas las funciones especificadas y sus respectivos tests. Aquí tienes los recursos bibliográficos y educativos que te serán útiles:

**📚 Bibliografía Principal**

**Tkinter (Interfaz Gráfica)**

1. **Real Python - Python GUI Programming With Tkinter**: Tutorial completo que cubre los conceptos básicos de programación GUI con Tkinter, incluyendo widgets, geometry managers y event handlers [Python GUI Programming With Tkinter – Real Python](https://realpython.com/python-gui-tkinter/)
   * URL: <https://realpython.com/python-gui-tkinter/>
2. **GeeksforGeeks - Python Tkinter Tutorial**: Biblioteca estándar GUI para Python que proporciona una forma rápida y fácil de crear aplicaciones de escritorio con widgets interactivos [GeeksforGeeks](https://www.geeksforgeeks.org/python/python-gui-tkinter/)[GeeksforGeeks](https://www.geeksforgeeks.org/python/python-tkinter-tutorial/)
   * URL: <https://www.geeksforgeeks.org/python/python-tkinter-tutorial/>
3. **Python GUIs - Tkinter Tutorial 2025**: Tutorial actualizado para crear aplicaciones GUI con Python y Qt [Tkinter Tutorial 2025, Create Python GUIs with TKinter](https://www.pythonguis.com/tkinter-tutorial/)
   * URL: <https://www.pythonguis.com/tkinter-tutorial/>
4. **Documentación Oficial de Python - tkinter**:
   * URL: <https://docs.python.org/3/library/tkinter.html>

**Pytest (Testing)**

1. **Real Python - Effective Python Testing With pytest**: Tutorial avanzado que cubre características intermedias y avanzadas de pytest como fixtures, marks, parameters y plugins [Effective Python Testing With pytest – Real Python](https://realpython.com/pytest-python-testing/)
   * URL: <https://realpython.com/pytest-python-testing/>
2. **TutorialsPoint - Pytest Tutorial**: Guía completa para dominar las pruebas en Python con pytest [Pytest Tutorial - Master Python Testing](https://www.tutorialspoint.com/pytest/index.htm)
   * URL: <https://www.tutorialspoint.com/pytest/index.htm>
3. **Better Stack - A Beginner's Guide to Unit Testing with Pytest**: Guía que explora los principios fundamentales de unit testing y las capacidades poderosas de Pytest [A Beginner's Guide to Unit Testing with Pytest | Better Stack Community](https://betterstack.com/community/guides/testing/pytest-guide/)
   * URL: <https://betterstack.com/community/guides/testing/pytest-guide/>
4. **DataCamp - How to Use Pytest for Unit Testing**: Tutorial práctico sobre pytest con funciones que tienen "test\_" como prefijo [How to Use Pytest for Unit Testing | DataCamp](https://www.datacamp.com/tutorial/pytest-tutorial-a-hands-on-guide-to-unit-testing)
   * URL: <https://www.datacamp.com/tutorial/pytest-tutorial-a-hands-on-guide-to-unit-testing>

**🎥 Videos Educativos Recomendados**

Para complementar tu aprendizaje, te sugiero buscar estos videos en YouTube:

1. **"Python Tkinter GUI Tutorial for Beginners"** - Corey Schafer
2. **"Python GUI Programming with Tkinter - Complete Course"** - freeCodeCamp
3. **"Pytest Tutorial - How to Test Python Code"** - Corey Schafer
4. **"Python Unit Testing With Pytest"** - Tech With Tim
5. **"Building Desktop Applications with Python and Tkinter"** - Real Python

**📖 Libros Recomendados**

1. **"Python GUI Programming with Tkinter"** - Alan D. Moore
2. **"Test-Driven Development with Python"** - Harry Percival
3. **"Clean Code: A Handbook of Agile Software Craftsmanship"** - Robert C. Martin
4. **"Python Tricks: The Book"** - Dan Bader

**🔗 Recursos Adicionales**

1. **Documentación oficial de datetime**: <https://docs.python.org/3/library/datetime.html>
2. **Python Enhancement Proposals (PEPs)**: <https://peps.python.org/>
3. **Stack Overflow**: Para resolver dudas específicas durante el desarrollo

**✅ Funciones Implementadas Según Tu Propuesta**

El código incluye exactamente las funciones que especificaste:

**Funciones Principales:**

* ✅ register\_arrival() - Registra llegadas de aeronaves
* ✅ assign\_to() - Asigna aeronaves a instalaciones (renombrada de assing\_to())
* ✅ check\_time\_used() - Calcula tiempo de uso de instalaciones
* ✅ check\_available() - Verifica instalaciones disponibles

**Funciones de Test:**

* ✅ test\_check\_time\_used() - Pruebas para cálculo de tiempo
* ✅ test\_check\_available() - Pruebas para verificación de disponibilidad
* ✅ Tests adicionales para todas las funciones principales

**Módulos Utilizados:**

* ✅ tkinter - Para la interfaz gráfica
* ✅ pytest - Para las pruebas unitarias
* ✅ datetime - Para manejo de fechas y tiempos