

## MiniProyecto

### Resumen (200 palabras):

Tarea	Tiempo estimado	Tiempo real
Formación	00:45h	1:00h
Traducción de Matlab a C++	3:00h	2:45h
Traducción de Matlab a Python	1:00h	1:15h

Problema	Solución
Funciones que en algunos lenguajes están implementadas y en otros no	Buscar su equivalencia en otros lenguajes o programarlas tú mismo (operaciones vectoriales, por ejemplo)
Funciones que devuelven más de un valor en C++	Añadir parámetros de salida
Comentarios en Matlab “%”	Equivalencia a comentario en C++ y Python
‘end’ genérico en Matlab para terminar condicionales y bucles.	Forma convencional de cerrar bucles en C++ y Python

### Lecciones aprendidas:

- Aprender a aprovechar las características comunes que tienen los lenguajes de programación para facilitar el aprendizaje de lenguajes que no conocemos.
- Importancia del propósito de un lenguaje: Matlab Orientado a cálculo numérico implementa funciones que no tenemos en lenguajes como C++.
- Relevancia del estilo del programador y de cómo influye cómo documente su código en su mantenimiento.
- Herramientas para aprender a generar test unitarios y de mantenimiento de código.
- Las estructuras condicionales no son demasiado útiles para usar test unitarios.
- Una función con el mismo nombre (atan2) puede tener usos distintos de un lenguaje a otro.
- Importancia de utilizar un lenguaje adecuado para tu propósito: Un mismo código con el lenguaje adecuado al fin de ese programa será mucho más sencillo que en uno que no lo es.

-Una vez identificadas las características generales entre el lenguaje de programación que desconozco y los que no, anotar las diferencias de código y documentarme sobre ellas.

-Analizar y conocer la magnitud del proyecto para poder inferir cuántas horas le vamos a dedicar.

-Diagrama de flujo para comprender el contexto del programa y cómo funciona. Sobre todo si son proyectos grandes.