

# **Software de obtención y procesamiento de datos a tiempo real con CUDA para sistema de interferometría**

## *Anexo I: Plan del Proyecto Software*

Proyecto de Fin de Carrera  
INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS



**VNiVERSiDAD  
D SALAMANCA**

Septiembre de 2012

Autor

**Álvaro Sánchez González**

Tutor

**Guillermo González Talaván**

Cotutor

**Francisco Valle Brozas**



## Contenido

<b>1. Introducción .....</b>	<b>5</b>
<b>2. Desarrollo del proyecto .....</b>	<b>7</b>
2.1. Fase de Inicio .....	8
2.2. Fase de Elaboración .....	8
2.2.1. Primera iteración: diseño del sistema base .....	8
2.2.2. Segunda iteración: diseño de la interfaz .....	9
2.2.3. Tercera iteración: refinamiento del diseño del sistema base .....	9
2.3. Fase de Construcción .....	9
2.3.1. Primera iteración: búsqueda del entorno de desarrollo .....	10
2.3.2. Segunda iteración: implementación de elementos del sistema base ....	10
2.3.3. Tercera iteración: puesta de los elementos en conjunto .....	11
2.3.4. Cuarta iteración: implementación de la funcionalidad añadida .....	11
2.4. Fase de Transición .....	12
2.5. Diagramas de Gantt .....	13
<b>3. Proceso de aprendizaje.....</b>	<b>17</b>
<b>4. Implicados en el proyecto.....</b>	<b>19</b>
<b>5. Viabilidad del proyecto.....</b>	<b>21</b>



## Índice de Figuras

Figura 1. Diagrama de Gantt: Fase de Inicio .....	13
Figura 2. Diagrama de Gantt: Fase de Elaboración .....	14
Figura 3. Diagrama de Gantt: Fase de Construcción .....	15
Figura 4. Diagrama de Gantt: Fase de Transición .....	16



## **1. Introducción**

Este primer anexo recoge toda la información de la planificación del proyecto.

Comienza con un resumen de la planificación, para pasar a una descripción más detallada de las distintas fases, incluyendo diagramas de Gantt para mostrar la distribución temporal.

Continúa con un breve resumen de los conocimientos que ha sido necesario adquirir, para la realización del proyecto.

Más adelante se puede encontrar una descripción de las personas implicadas en el proyecto: los tutores.

Y por último hace un análisis de viabilidad, corto, pero con detalles importantes acerca de los aspectos económicos y técnicos que han sido necesarios para el proyecto.





## 2. Desarrollo del proyecto

El desarrollo ha seguido el Proceso Unificado. Esto ha sido posible gracias a una de las grandes ventajas del Proceso Unificado: la versatilidad que tiene a la hora de planificar un proyecto debido al uso de iteraciones, permitiendo contener todo tipo de modelos de proceso software.

En este caso, dadas las características del proyecto, se optó por un modelo bastante lineal durante la etapa de ingeniería: una toma de requisitos lo más completa posible, así como un diseño lo más robusto posible (que soportara cambios en los requisitos); para pasar a una etapa de construcción de módulos, muy incremental y basada en prototipos para realizar las pruebas unitarias, así como en otros prototipos para la realización de pruebas de integración entre ellos.

Al final de todo este proceso, el único paso restante fue construir el programa principal juntando todos los módulos, y por tanto la funcionalidad, en un solo proyecto, y pasar a las pruebas finales y corrección de pequeños fallos no detectados con las pruebas anteriores.

Se eligió este tipo de modelo, lineal durante la toma de requisitos, y más iterativo durante la implementación, debido a la naturaleza temporal del proyecto. Desde el principio, Marzo de 2012, se quiso poder estimar si el proyecto se iba a poder entregar para Septiembre de 2012, sin embargo, una escueta toma de requisitos al principio hubiera hecho muy difícil poder realizar esa estimación.

Por otra parte, dado que el proyecto utiliza distintas herramientas muy diferenciadas (Cámara, CUDA, representación de gráficos 2D, representación de gráficos 3D,...) era absolutamente imprescindible tener clases muy consistentes y bien diseñadas, con poco acoplamiento, pero mucha cohesión, que integraran cada uno de estos aspectos, resultado que se logró perfectamente mediante el desarrollo incremental utilizado.

## **2.1. Fase de Inicio**

Esta etapa se centró en la toma de requisitos.

Primeramente se hizo una toma de requisitos más superficial de cara a la realización de la propuesta del proyecto.

Después de esto, comenzó la verdadera toma de requisitos, con una entrevista con el cliente, que en este caso coincide con el cotutor, así como entrevistas posteriores de refinamiento de estos requisitos, según se iban extrayendo los casos de uso. El objetivo de salir de esta fase con una completa especificación de requisitos se vio cumplido, aunque más adelante fueron necesarias ligeras variaciones, y la inclusión de algún requisito adicional.

## **2.2. Fase de Elaboración**

En esta fase se finalizó el análisis y se realizó la mayor parte del diseño del sistema. Dado que uno de los requisitos era el trabajo con CUDA, se realizaron también pruebas externas al proyecto, para conocer mejor el funcionamiento de esa herramienta y poder realizar un diseño mejor a la hora de incluirlo.

Esta fase ha consistido en tres iteraciones:

### **2.2.1. Primera Iteración: diseño del sistema base**

La primera interacción se centra en el diseño de las clases fundamentales del sistema, que contienen por una parte la funcionalidad del programa y por otra parte las estructuras de datos que se van a transmitir de unos elementos a otros del programa.

Fue la primera vez que se comenzó a trabajar en el dominio de la solución, buscando técnicas para obtener una buena abstracción en el uso de la cámara, y especialmente, en el uso de CUDA o la CPU, de forma que esta doble implementación fuera totalmente transparente a la mayor parte de módulos del programa. Para conseguir

eso, se recurrió al uso de distintos patrones, explicados con más detalle en el Anexo III de Diseño.

### 2.2.2. Segunda iteración: diseño de la interfaz

Una vez diseñada la parte central del programa se realizaron unos bocetos sobre papel con la estructura de la interfaz y las distintas ventanas y diálogos que iban a ser necesarios.

Este diseño se enseñó al cliente, que aportó alguna idea adicional para la interfaz.

No obstante esta interfaz no se implementó hasta ya muy avanzada la fase de construcción, pues para las pruebas de los diferentes productos de las iteraciones base fue mucho más útil el uso de interfaces “de usar y tirar”, simples, con únicamente los elementos necesarios para la prueba realizada.

Por otra parte se buscaron las bibliotecas necesarias para la representación gráfica, y se diseñaron las clases que permitieran representar las estructuras de datos diseñadas en la iteración anterior.

### 2.2.3. Tercera iteración: refinamiento del diseño del sistema base

El diseño de la interfaz ayudó por otra parte a mejorar la concepción del proyecto aportando una mejor visión global del sistema. Esto hizo que aparecieran ideas nuevas acerca de cómo organizar el sistema base, por lo que tras el diseño de la interfaz, se produjo un importante refinamiento del diseño alcanzado en la primera fase.

## **2.3. Fase de Construcción**

Esta fase se plantea de forma parecida al diseño, pero incluyendo esta vez muchas más iteraciones en lo que a implementación-pruebas se refiere, prácticamente una iteración para cada una de los módulos, que se agrupan en cuatro iteraciones principales.

### 2.3.1. Primera iteración: búsqueda del entorno de desarrollo

Antes de comenzar a implementar, se buscó un entorno de desarrollo que fuera adecuado, de acuerdo a dos objetivos:

- Desde el principio se planteó el funcionamiento del programa en sistemas Windows, dada su popularidad. Sin embargo, no se quiso limitar a que en un futuro el programa pudiera funcionar en otros sistemas operativos.
- Además, el uso de CUDA a bajo nivel, exige código en C, por lo que el proyecto se implementaría en C/C++.

Estas dos condiciones llevaron a pensar en la biblioteca Qt y su entorno de desarrollo Qt Creator. Sin embargo, CUDA, bajo el sistema Windows sólo da soporte oficial a Microsoft Visual Studio (Dado que requiere una etapa adicional en la compilación con un compilador externo, y a continuación un enlazado correcto), por lo que no se sabía si iba a ser posible el uso de Qt.

Tras varios días de pruebas, se consiguió integrar CUDA con Qt, y por lo tanto se fijó C/C++ y la biblioteca Qt junto con Qt Creator como entorno de desarrollo.

Una vez determinado eso, se compilaron todas las bibliotecas necesarias hasta el momento para comprobar su correcto funcionamiento en ese entorno.

Nótese que durante esta iteración hubo una parada del proyecto, con motivo de la preparación de exámenes para la convocatoria de Mayo.

### 2.3.2. Segunda iteración: implementación de elementos del sistema base

En esta iteración se implementaron los módulos principales, y se realizaron pruebas individuales para cada uno de estos elementos. Esta implementación tuvo lugar en la en el siguiente orden lógico, de manera que fuera posible la realización de pruebas:

- Implementación de las estructuras de datos que contendrían la información candidata a ser representada como resultado de procesamiento.

- Implementación parcial de las clases que permitirían representar gráficamente las estructuras de datos antes implementadas, en distintos formatos, para su uso en las interfaces de “usar y tirar”.
- Implementación de las clases con la funcionalidad y los algoritmos para realizar los cálculos exigidos por los requisitos sobre las imágenes, por ahora cargadas desde archivo. En un primer momento sólo se implementaron las clases que realizaban estos cálculos usando CUDA.
- Implementación de la cámara y las clases alrededor de la cámara.
- Mejora de la implementación de las clases de representación gráfica, que en un primer momento sólo se implementaron lo justo y necesario para poder representar en ellas.

### 2.3.3. Tercera iteración: puesta de los elementos en conjunto

Cómo indica el título, esta iteración consistió en conectar todos los elementos que se habían implementado anteriormente, a través de un controlador principal, y una interfaz que permitiera acceder a la funcionalidad implementada, e incluso, a la funcionalidad que todavía carecía de implementación.

Hecho esto, se pudieron realizar las pruebas de manejo del programa y hacer una primera evaluación de la usabilidad de la interfaz.

Además se enseñó al cliente y al tutor, como un esqueleto de lo que más adelante sería el programa definitivo, al que éstos aportarían modificaciones y nuevas ideas.

### 2.3.4. Cuarta iteración: implementación de la funcionalidad añadida

Para esta última iteración sólo se dejaron tres bloques de funcionalidad:

- Modo de funcionamiento “en vivo”: Uno de los requisitos más importantes del proyecto era la posibilidad de funcionamiento a en tiempo real, con el ordenador conectado a una cámara en el laboratorio. Para esto, dado que se había realizado un buen diseño de los elementos base, sólo fue necesario implementar una clase controlador que dirigiera el proceso de adquisición y

procesamiento de datos periódicamente, así como una interfaz específica para la representación a tiempo real.

- Funcionalidad adicional de guardado y exportación de elementos: Esto es, las clases y métodos que permitieran guardar de forma persistente los resultados de procesado, listas de algoritmos, máscaras,... Además apareció un requisito de última hora, propuesto por parte del tutor: la capacidad del programa de cargar y guardar proyectos completos, que en un principio no se había planteado. No obstante no hubo ningún problema en añadirlo a la lista de requisitos, y realizar su diseño e implementación.
- Además, sin demasiada dificultad, debido a la buena abstracción usada en el diseño de la cámara, se implementó la posibilidad de uso de un simulador de cámara, para poder contemplar toda la funcionalidad del programa aún careciendo de acceso a la cámara y al instrumental del experimento para el que se diseñó el programa.

## ***2.4. Fase de Transición***

Al inicio de esta fase se cuenta con una versión beta del proyecto. El programa ya se considera un sistema con entidad, sin embargo todavía necesita de la realización de muchas pruebas, que antes no se habían podido realizar dada la falta de conexión entre los módulos del programa. Estas pruebas incluyen pruebas de simulación, de interfaz, de forzado de errores (Para comprobar los mecanismos de defensa del programa), de guardado y carga de elementos, y en general de funcionamiento de usuario del programa. En esta última parte además se realizaron pruebas en el laboratorio del funcionamiento de la aplicación a tiempo real con resultado exitoso.

Además, en esta etapa del desarrollo se aprovechó para incluir en el programa elementos secundarios de facilidad de uso y elegancia del programa, como son alertas de guardado, opciones de ajuste de los gráficos, almacenamiento de información de configuración persistente de una ejecución a otra, y por supuesto la creación de un instalador.

## 2.5. Diagramas de Gantt

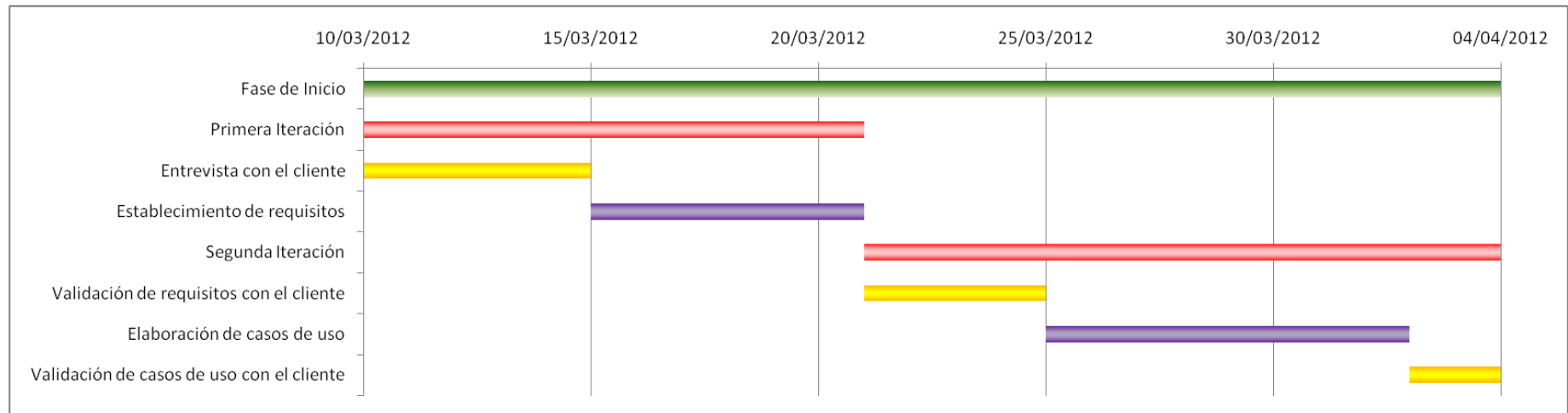


Figura 1. Diagrama de Gantt: Fase de Inicio

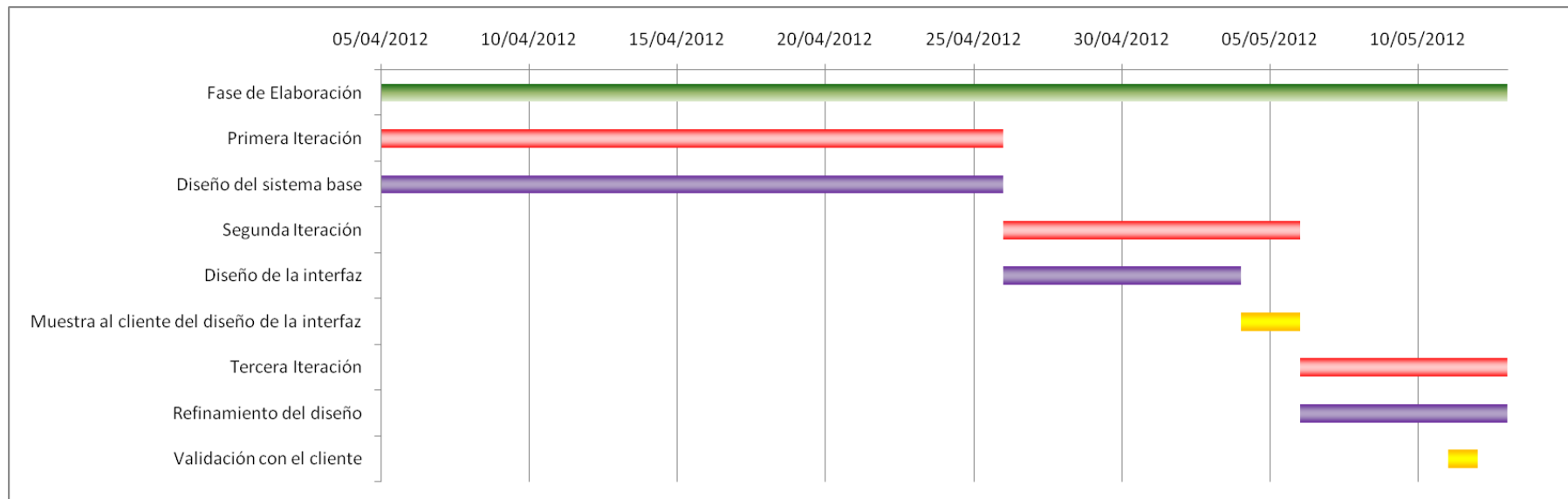


Figura 2. Diagrama de Gantt: Fase de Elaboración



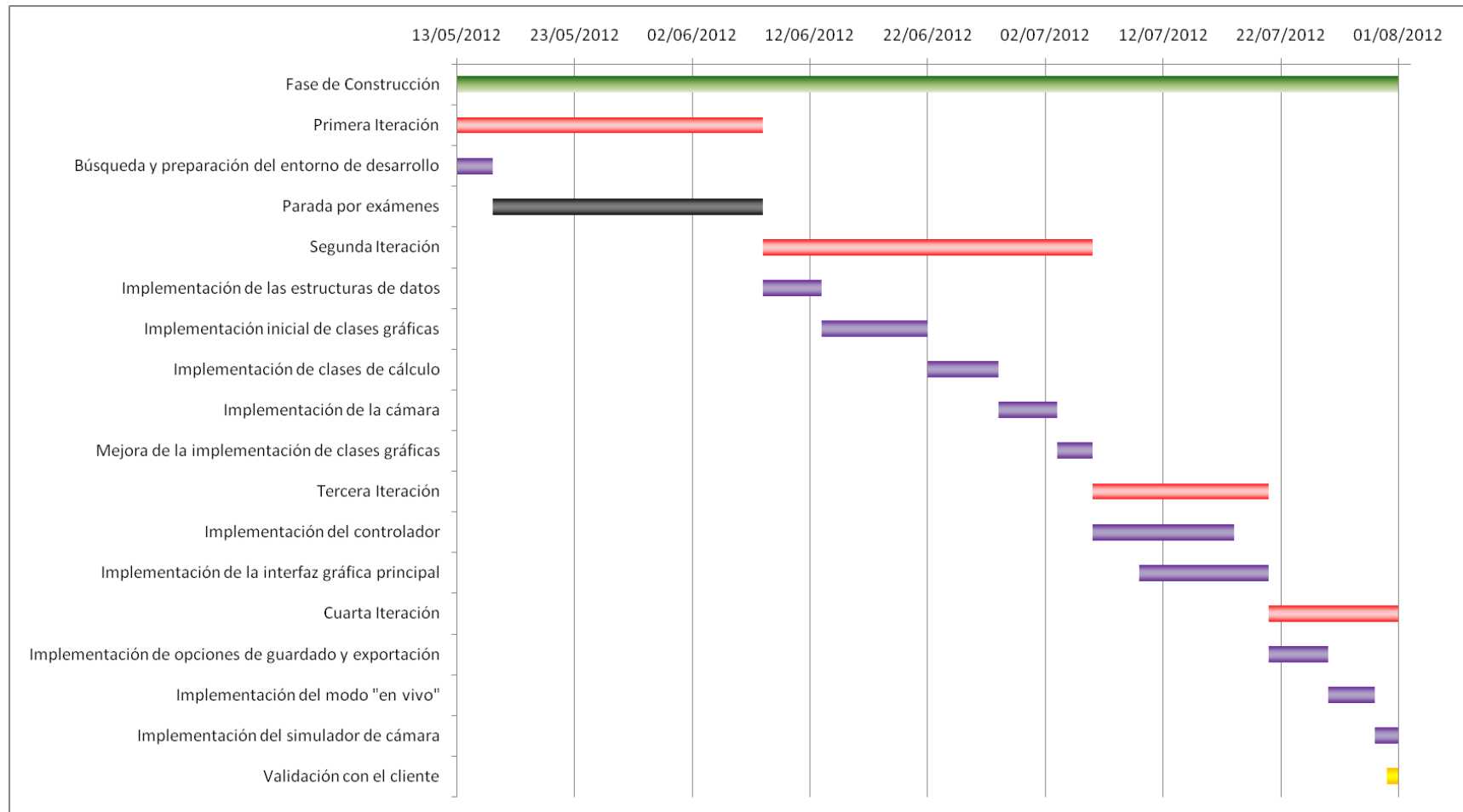


Figura 3. Diagrama de Gantt: Fase de Construcción

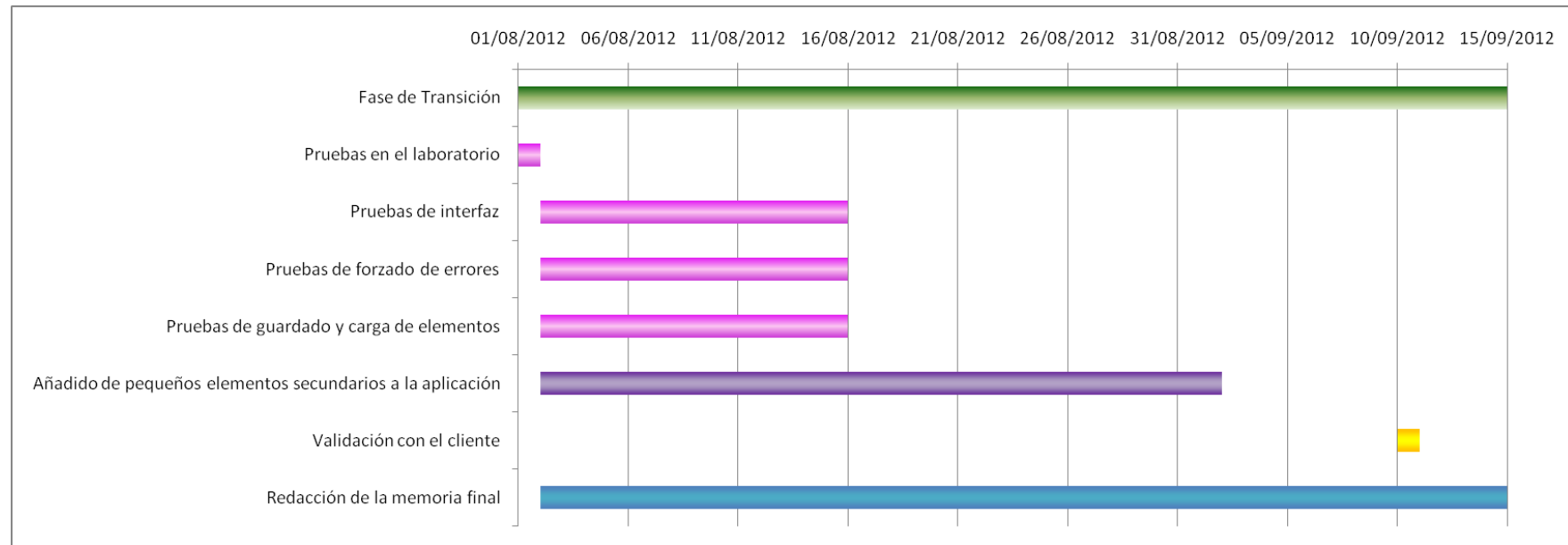


Figura 4. Diagrama de Gantt: Fase de Transición

### 3. Proceso de aprendizaje

Para realizar el proyecto ha sido necesario el aprendizaje de uso de diferentes herramientas y sistemas, además de la propia formación en la temática del proyecto: la obtención y el procesado de la fase.

De forma previa al proyecto consistió en principalmente dos elementos:

- Por una parte ha sido necesario el aprendizaje de uso de C++ como lenguaje de programación orientado a objetos. Este aprendizaje ha sido gradual, con distintos fuentes en la red.
- Por otra parte ha sido fundamental el aprendizaje y entrenamiento en CUDA. Para ellos se utilizó principalmente el libro “CUDA by example” de Jason Sanders y Edward Kandrot, un libro cuya experiencia de lectura ha sido más que satisfactoria y que, como opinión personal, es una forma excelente de acercamiento al mundo de CUDA.

Después, ya durante la implementación del sistema:

- Aprendizaje de uso de las clases de Qt necesarias, especialmente de dos tipos: clases de implementación de controles gráficos, y clases para posibilitar ejecución multi-hilo.
- Aprendizaje de uso de las bibliotecas de representación gráfica QwtPlot y QwtPlot3D, ambas multiplataforma.
- Aprendizaje de uso de la API  $\mu$ Eye, para las cámaras utilizadas, también multiplataforma.
- Aprendizaje de uso de software externo para exportado de vídeos (MEncoder) y de animaciones (ImageMagick) ambos multiplataforma.
- Aprendizaje de uso de la biblioteca multiplataforma QuaZIP, para el guardado de proyectos (Un proyecto consta de varios archivos que se comprimen en un fichero ZIP, con extensión “.jpp”).



## **4. Implicados en el proyecto**

Además del alumno que realiza el proyecto hubo dos implicados adicionales con los que se realizaron diversas entrevistas.

### **- Entrevistas con el tutor Guillermo González Talaván**

El rol de Guillermo ha sido el de tutor principal, supervisando las distintas fases del proyecto, así como los hitos planteados para cada una de ellas, desde un punto de vista más informático, aunque aportando también ideas a la parte física del proyecto, dada su condición de Licenciado en Física.

Las reuniones con él se han realizado en los momentos clave entre fases del proyecto en las que se proponían modificaciones al diseño o a la interfaz, y se indicaban los objetivos para la siguiente fase.

### **- Entrevistas con el cotutor y cliente Francisco Valle Brozas**

Francisco ha tenido un doble papel en el proyecto: por una parte como cotutor del proyecto y por otra parte como cliente ya que la idea inicial para el proyecto era suya y por tanto era uno de los principales interesados en el resultado.

Como cliente, se tuvieron con él todas las entrevistas de toma de requisitos y de validación de requisitos, así como de aceptación de los módulos que se iban implementando.

Como cotutor, se encargó de la formación del alumno en la mayor parte de la materia teórica relacionada el procedimiento de procesamiento de las imágenes, así como de la resolución de dudas que surgían por parte del mismo alumno y de realizar una revisión de la memoria y el resumen al finalizar el proyecto.

Además, él fue el encargado de poner a punto el experimento sobre el que se probaría el sistema en tiempo real y de proporcionar las imágenes experimentales necesarias

durante todo el proceso de construcción del programa para las pruebas del software funcionando por separado del montaje experimental.

## 5. Viabilidad del proyecto

▪ **Viabilidad económica:** el departamento de física aplicada se ha preocupado por proporcionar un ordenador personal, un espacio de trabajo para la realización del proyecto y una tarjeta gráfica compatible con CUDA que cumpliera con las especificaciones necesarias para su uso con el programa. El resto de herramientas y bibliotecas software usadas son libres o gratuitas.

▪ **Viabilidad técnica:** el material para el experimento de interferometría necesario ya estaba disponible en el laboratorio, y puesto a punto para la utilización de un sistema como el creado, por lo que no se encontró ninguna dificultad técnica para su uso.

▪ **Viabilidad legal:** el proyecto ha usado software libre de código abierto o bien software con permiso explícito para aplicaciones tanto libres como propietarias, por lo que el futuro del software sólo depende de los objetivos del Área de Óptica del departamento de Física Aplicada, bajo el cual se ha realizado y del que se ha obtenido la financiación, en conjunto con el propio autor del proyecto.