# Análisis

Este apartado de la memoria engloba todos los aspectos relacionados con el análisis del proyecto software. Primero, se definirán los casos de uso para representar los distintos escenarios que pueden presentarse en el sistema; en segundo lugar, se especificarán los requisitos del proyecto derivados de esos casos de uso; en tercer lugar, como resultado se llevará a cabo una planificación temporal para esos requisitos y, por último, se detallarán los costes del proyecto.

## Casos de Uso

Para el desarrollo de esta aplicación se hizo un estudio en alumnos de la asignatura de Programación de Aplicaciones Gráficas como también en gente que trabaja en el sector de la creación de software, definiéndose unos casos de uso.

Los casos de uso ayudan a la hora de poder especificar los requisitos necesarios para el desarrollo del software. Según Pressman [1], los casos de uso se describen como:

“En esencia, un caso de uso cuenta una historia estilizada sobre cómo un usuario final (que desempeña uno de varios roles posibles) interactúa con el sistema bajo un conjunto específico de circunstancias.”

A continuación, se muestran los principales casos de uso, que más tarde se describirán de forma abstracta conjuntamente con su diagrama de caso de uso. En este sistema se pueden especificar tres casos de uso:

* **Creación y guardado de un shader GLSL**: Es el caso más genérico de todos, ya que representa la secuencia de interactuar con todas las partes esenciales de la aplicación.
* **Carga de un shader GLSL**: Es un caso parecido al anterior, pero se puede discernir en que no se escribe un shader, sino que se parte de uno ya existente.
* **Elegir idioma:** Representa el proceso por el cual el usuario cambia el idioma del sistema.
* **Interacción con la cámara**: Se muestra el escenario en el que usuario modifica el estado actual de la cámara y afecta al espacio 3D generado.

### Creación y guardado de un shader GLSL

|  |  |
| --- | --- |
| Caso de Uso | Creación y guardado de un shader GLSL |
| Actor Principal | Usuario |
| Sistema | Toulouse Compiler[[1]](#footnote-1) |
| Precondiciones | Tener la aplicación instalada en el sistema |
| Flujo Básico |  |
| 1 | Escribe un shader |
| 2 | Selecciona el tipo de **Index Buffer Object (IBO)**  a usar |
| 3 | Selecciona el tipo de objeto a usar |
| 4 | Ingresa los Uniforms necesarios |
| 5 | El usuario pulsa el botón de compilar |
| 6 | El usuario visualiza el resultado del shader |
| 7 | El usuario guarda el shader |
| Escenarios Alternativos |  |
| 6.1 | El shader es erróneo y solo visualiza los tipos de errores que hay en el shader |

Tabla Caso de uso(escribir shader)

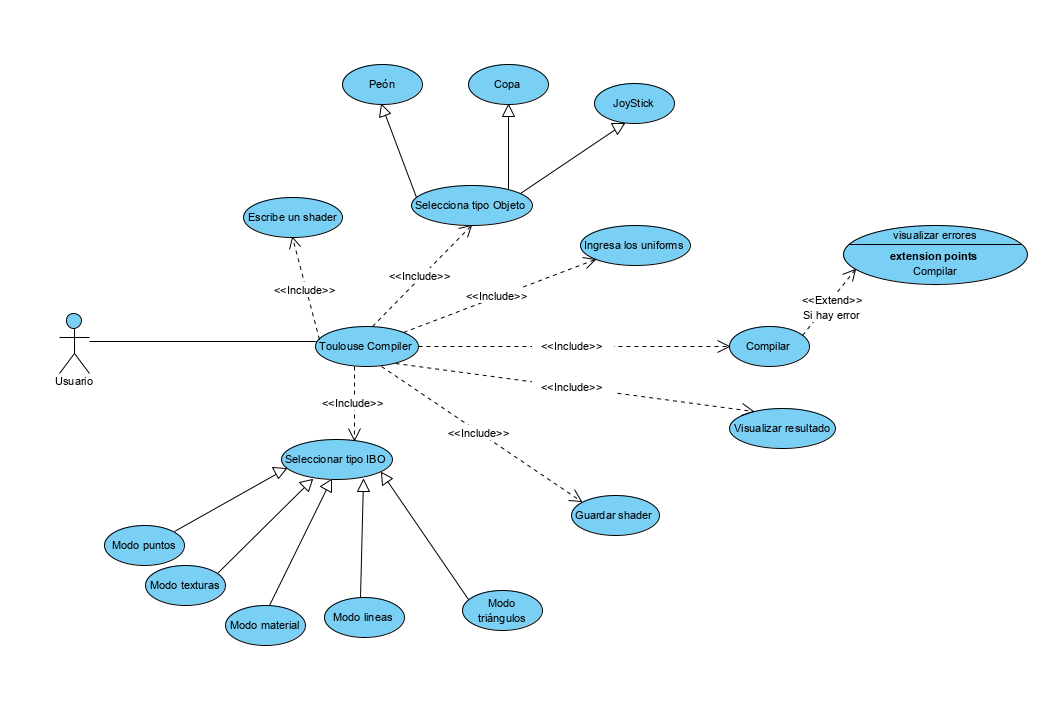


Ilustración Diagrama casos de uso (escribir shader)

### Carga de un shader GLSL

|  |  |
| --- | --- |
| Caso de Uso | Carga de un shader GLSL |
| Actor Principal | Usuario |
| Sistema | Toulouse Compiler |
| Precondiciones | Tener la aplicación instalada en el sistema |
| Flujo Básico |  |
| 1 | Abre un shader existente |
| 2 | Selecciona el tipo de **Index Buffer Object (IBO)**  a usar |
| 3 | Selecciona el tipo de objeto a usar |
| 4 | Ingresa los Uniforms necesarios |
| 5 | El usuario pulsa el botón de compilar |
| 6 | El usuario visualiza el resultado del shader |
| 7 | El usuario guarda el shader |
| Escenarios Alternativos |  |
| 1.1 | No se encuentra el archivo buscado |
| 6.1 | El shader es erróneo y solo visualiza los tipos de errores que hay en el shader |

Tabla Caso de uso (cargar shaders)

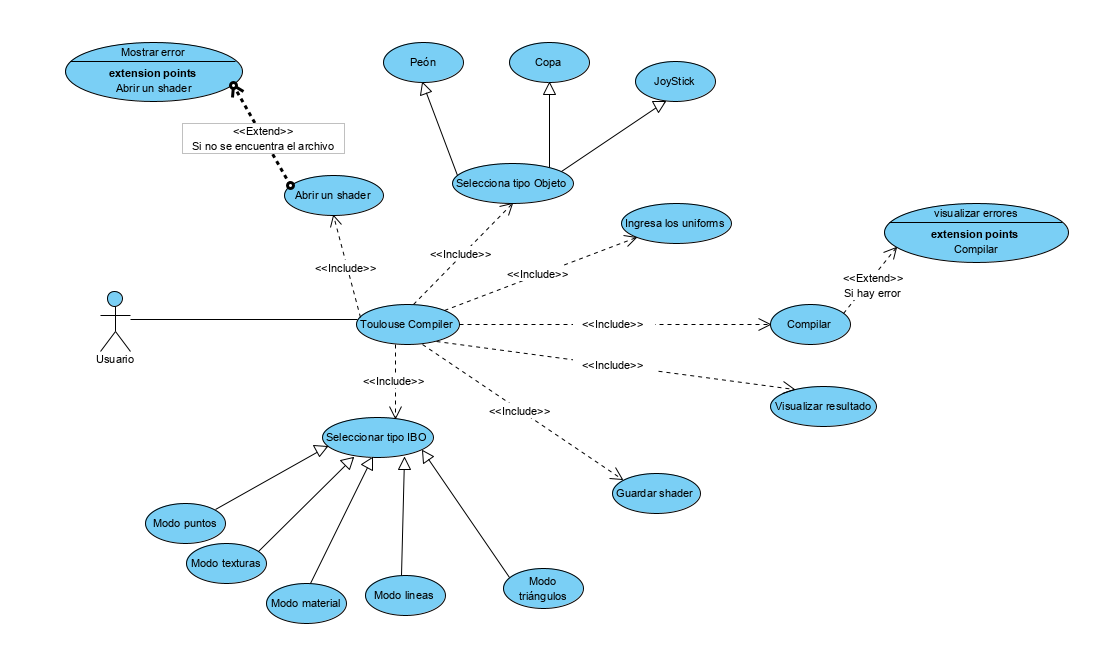


Ilustración 2 Diagrama caso de uso (cargar shader)

### Elegir idioma

|  |  |
| --- | --- |
| Caso de Uso | Elegir idioma |
| Actor Principal | Usuario |
| Sistema | Toulouse Compiler |
| Precondiciones | Tener la aplicación instalada en el sistema |
| Flujo Básico |  |
| 1 | El usuario pulsa el botón de idioma |
| 2 | Aplica un idioma |

Tabla Caso de uso (cambiar idioma)

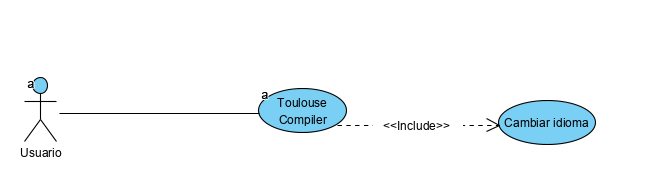


Ilustración 3 Diagrama caso de uso (cambiar idioma)

### Interacción con la cámara

## Requisitos

Tras analizar los casos de uso, es necesario especificar los requisitos tanto funcionales como los no funcionales que tiene que cumplir la aplicación para el correcto desarrollo.

### Requisitos funcionales

Los requisitos funcionales representan las funcionalidades que el sistema debe ser capaz de realizar al final del desarrollo. A continuación, se enumeran los requisitos:

* Se debe tener una consola para mostrar los errores tantos de los shaders como del funcionamiento de la aplicación.
* Se deben visualizar objetos en 3d para el posterior tratamiento con Shaders.
* Se debe poder cargar y guardar Shaders.
* Se tiene que poder cambiar el tipo de IBO a usar.

### Requisitos no funcionales

Una vez definidos los requisitos funcionales se definirán los no funcionales. Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema, sino a las propiedades del sistema: rendimiento, interfaz, seguridad, usabilidad, seguridad, etc.

* La parte donde se escribe el código de los shaders, se verá afectada por el efecto de resaltado de las palabras reservadas (se destacan aquellas variables reconocidas por el sistema como palabras reservadas de GLSL).
* La interfaz será intuitiva, seguirá una guía de estilo uniforme, manteniendo la misma tipografía, …
* La interfaz del programa deberá tener dos idiomas, español e inglés, siendo posible añadir otros idiomas más delante de forma sencilla.
* Se podrá añadir la opción de cargar objetos del usuario de forma sencilla.
* La aplicación debe ser estable.

## Planificación temporal

La planificación temporal de un proyecto software evoluciona con el tiempo y permite identificar, definir y planificar las actividades específicas para realizar una tarea.

La precisión en la planificación puede a veces ser más importante que la precisión en el cálculo de los costes ya que los costos añadidos pueden ser absorbidos por la inflación o amortización. Sin embargo, un error de precisión en la planificación puede ocasionar clientes insatisfechos, aumentar los costes internos por problemas adicionales, etc.

Ya que el presente proyecto es un Trabajo de Fin de Grado, los cálculos de tamaño del proyecto están supeditados el tiempo disponible. En cuanto al esfuerzo, se dispone de tan un solo efectivo (la persona autora del trabajo). Resumiéndose el tiempo disponible en 3 meses, 6 horas diarias, en total aproximadamente 300 horas. Considerando el último de ellos como medio mes para reuniones con los tutores especificando las posibles modificaciones resultantes de las reuniones. Y el otro medio para el testeo final y refinamiento del programa.

Para la parte de la planificación temporal del desarrollo del proyecto, se generó un diagrama de Gantt, el cual especificaba el tiempo necesario para cada una de las partes del desarrollo y cuya especificación temporal es la siguiente: (*ver ilustración 1.10.1*):

* Búsqueda bibliográfica. 4 días.
* Estudio del entorno de desarrollo QT. 4 días.
  + Creación de programa simple funcional con QT. 1 día.
  + Creación más compleja de aplicación con OpenGL. 3 días.
* Creación del primer prototipo.15 días.
  + Integración de los shader programs al prototipo.2 días.
  + Integración de las prácticas de PAG.8 días.
  + Implementación del Code Editor. 5 días.
* Diseño de la interfaz.7 días.
  + Diseño e Implementación de Tabla Uniforms.3 días.
  + Diseño e Implementación del Log.2 días.
  + Integración del Code Editor a la interfaz. 2 días.
* Implementación de utilidades del programa. 5 días
  + Controles de cámara para OpenGL. 2 días.
  + Opciones selección distintos objetos. 3 días.
* Extensión de utilidades. 5 días.
  + Subrayado del código.1 día.
  + cargar y guardar shaders.1 día
  + Selección idioma. 3 días.
* Refinamiento de la aplicación. 1 día.
* Documentación (Se realiza de forma transversal al proyecto).50 días

Destacando que los objetivos a realizar dentro de la tarea asignada se planearan con un diagrama de Kanban (*Ver ilustración 1.10.2*).

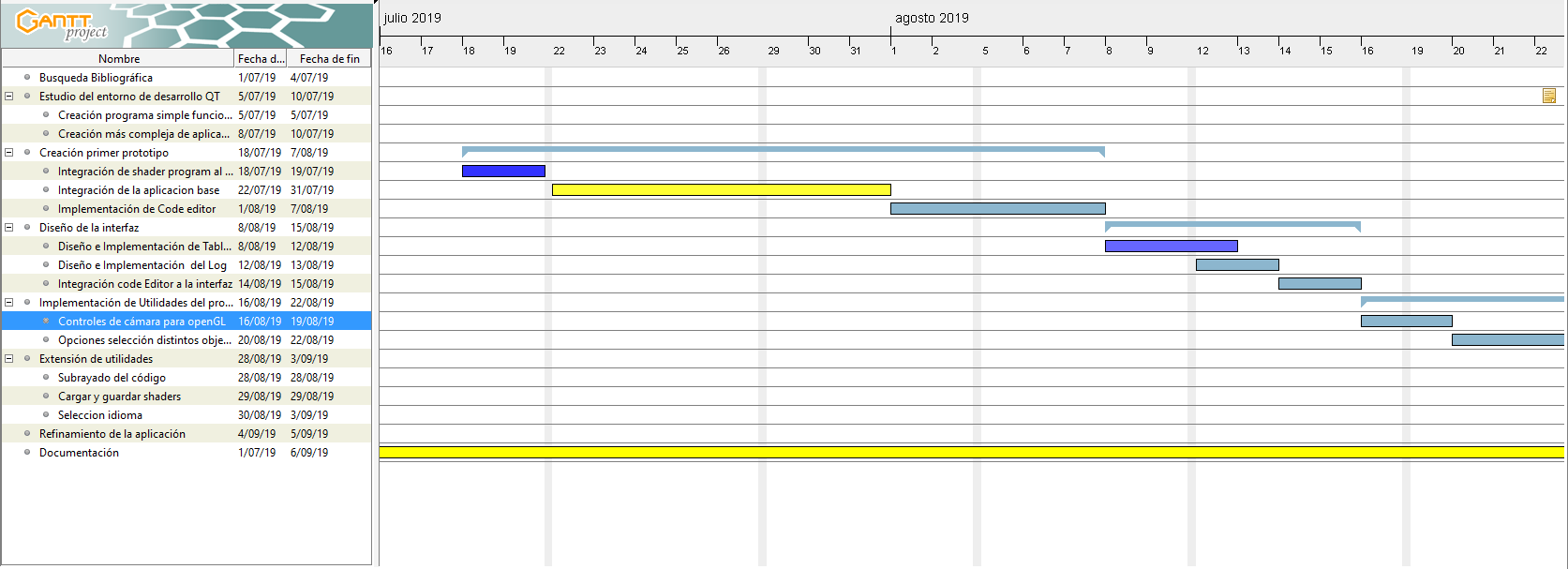


Ilustración 4 Diagrama de Gantt del Proyecto

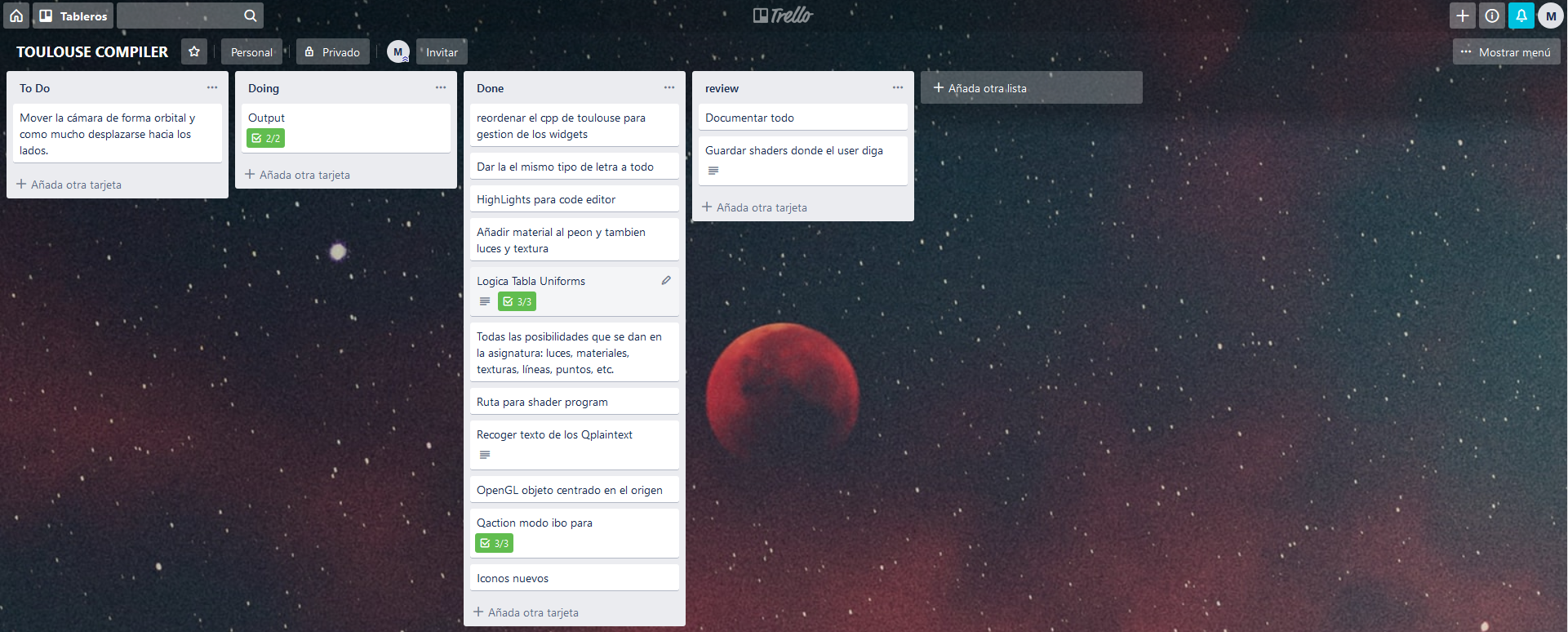


Ilustración 5 Diagrama de Kanban del Proyecto

## Metodología de desarrollo de software

Para el desarrollo de este proyecto se seguirá una metodología **SCRUM,** siguiendo las bases marcadas en *The Scrum Guide* [2] de los creadores de dicha metodología. Con esta metodología se obtiene una gran flexibilidad y adaptación a las necesidades del cliente, se consiguen resultados anticipados, se gestiona regularmente las expectativas del cliente, … (en este caso los clientes son los tutores).

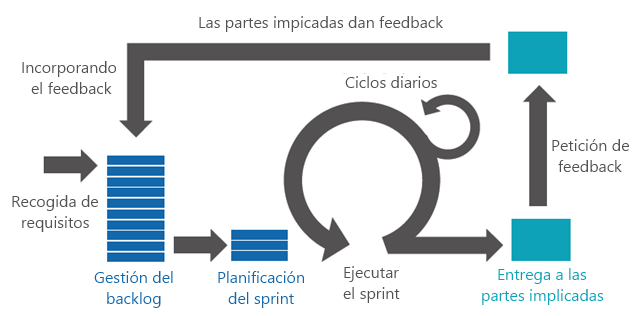


Ilustración 4.4.6 Diagrama fases SCRUM

Se necesitará de varias iteraciones dentro de la metodología para llegar a la fase final del proyecto.

A continuación, se explicará las fases dentro de la metodología SCRUM.

### Recogida de requisitos

El proceso comienza con la generación de la lista de objetivos o requisitos priorizada, que actúa como plan del proyecto y que es entregada por el cliente al equipo de desarrollo.

La lista de requisitos priorizada representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto.

### Gestión de backlog

Después de que se haya revisado la representación de los requisitos, para cada requisito se indica la importancia y el costo estimado de completarlo, que en este caso será tiempo.

### Sprint Planning Meeting

Un **Sprint** es una unidad de trabajo que agrupa un conjunto de tareas en un periodo de tiempo. La primera iteración es de planificación y está compuesta por dos partes:

* **Selección de requisitos:** Es la iteración entre cliente y equipo, el momento en que el equipo pregunta al cliente las dudas que surgen y se seleccionan los requisitos más prioritarios que se comprometen a completar en la iteración.
* **Planificación de la iteración:** Se elabora la lista de tareas o acciones necesarias para desarrollar los requisitos a los que se han comprometido.

De esta forma siempre se mantiene el contacto con el cliente y este está más involucrado en el proyecto.

### Ejecución de sprint

En la metodología SCRUM un proyecto se ejecuta en bloques temporales cortos y fijos, llamados sprint, que son iteraciones de 2 semanas. Si se sobrepasa este tiempo, como máximo un sprint puede tomar 4 semanas.

De esta forma no se gasta más tiempo en una parte del proyecto y en otras menos.

### Inspección e iteración

El último día de la iteración se realiza la reunión de revisión de la iteración, y se compone de dos partes:

**Sprint Review:** El equipo desarrollador presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado. El cliente revisa el entregable y se adaptan las mejoras necesarias.

**Sprint Retrospective:** En esta fase el equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, enfocando el proceso a la mejora continua del equipo.

De esta forma se puede seguir evolucionando la aplicación y el equipo va mejorando con el desarrollo.

## Presupuesto

Una parte importante del proyecto es cuanto se está dispuesto a pagar para el desarrollo del producto, es decir el presupuesto del que se dispondrá. Se supondrá que estos gastos los asume una empresa ficticia, por lo tanto, no se dedicará únicamente a este proyecto. Esta suposición hará posible que se puedan repartir los recursos entre estos proyectos (se pensara que están desarrollando 4 proyectos).

Para poder especificar correctamente los costes hay que diferenciarlos en:

* Costes de materiales.
* Costes de software.
* Costes de hardware.
* Costes de personal.

Todos los costes llevan aplicado el 21% de los impuestos.

### Costes de materiales

Los costes de carácter material, son recursos de uso general y coste reducido y donde no va incluido los costes de hardware para el desarrollo. En la tabla X se muestran los costes materiales:

|  |  |  |
| --- | --- | --- |
| Nombre | Objetivo | Coste(€) |
| Post-It 47.6 mm x 47.6 mm | Destinado a señalar las tareas que se realizaran ese día y errores detectados. | 3,00 € |
| Caja de 50 unidades de bolis BIC | Para diseños en papel y apuntar tareas | 11,28 € |
| Coste Total | | 14,28 € |

Tabla Desglose de costes en materiales

### Costes de Software

En este apartado se incluirán todos los programas utilizados en el proyecto. La mayoría del software es gratuito o se abona mensualmente, así que se tendrá en cuenta y no se amortizaran.

|  |  |  |
| --- | --- | --- |
| Nombre | Objetivo | Coste(€) |
| Adobe Photoshop CC | Destinado a la generación de los partes de la interfaz | 24,19 €/mes x 3 |
| Office 365 | Realización de la documentación | 7,00 €/mes x 3 |
| Visual Studio 2017 versión estudiante | Compilación y generación de código | 0 € |
| Github | Control de versiones | 0 € |
| Trello | Generador de diagramas Kanban | 0 € |
| QT Creator | Entorno para la generación de aplicaciones gráficas | 0 € |
| Coste Total (€) | | 101,15 € |

Tabla Desglose de costes de software.

Los programas mostrados serán necesarios para el desarrollo del programa, pero puede darse el caso que se necesite de algún programa externo que no sea gratuito, que ira dentro del porcentaje adicional para costes imprevistos.

### Costes de Hardware

Estos costes engloban los equipos necesarios u otros dispositivos necesarios para el desarrollo del proyecto. Al ser solo una persona la que está trabajando en él, solo es necesario la compra de un ordenador.

Los equipos informáticos son amortizables según el tiempo de uso, ya que no solo se usa para un solo proyecto, sino que se pueden usar para varias cosas en la empresa. Suele amortizarse con un mínimo de 4 años y un máximo de 8 años. Según la fórmula de la amortización [3]:

Vo: Valor inicial del activo.

Vr: Valor residual del activo al final de su vida útil.

n: Número de años de vida útil del activo.

Se estima que se necesitará un ordenador cuyo valor inicial serán 1000 €, para el desarrollo de este proyecto. Se supondrá que la vida útil del equipo será esos 4 años (35.040 horas) y el valor residual del activo al final de su vida será 0 €, teniendo en cuenta estos datos la amortización del equipo serían 0.0285…€ la hora de funcionamiento del equipo.

Como el equipo se usará durante 3 meses para este proyecto, la amortización final son 63.70 € (2.232 horas en 3 meses).

### Coste de Personal

En este apartado se mostrarán los costes asociados a la parte de personal. Toca destacar que todo el trabajo de este proyecto recae sobre solo un alumno, por lo tanto, hará distintos roles dentro del desarrollo del proyecto.

El salario base de los trabajadores se ha extraído del *Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública.* [4]

Para la realización del proyecto se necesitará de los siguientes roles:

* **Analista/Diseñador**: Se encargará de todo el análisis y diseño del sistema.
* **Programador**: Se encargará de la implementación del sistema.
* **Supervisor de pruebas**: Encargado de la experimentación y pruebas de usuario.
* **Jefe del proyecto**: Se encargará de supervisar, generar la documentación y difundir los resultados obtenidos.

Al sueldo mensual de los trabajadores hay que aplicarle el 33% de impuestos a la empresa en concepto de gasto para la Seguridad Social.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nombre | Sueldo base mensual | Retención mensual | Meses | Dedicación al proyecto | Coste(€) |
| Analista/diseñador | 1.535,22 € | 506,62 € | 2 | 25% | 1.020,92 € |
| Programador | 1.089,20 € | 359.43 € | 3 | 100% | 4.945,89 € |
| Supervisor de pruebas | 828,39 € | 273,36 € | 3 | 25% | 826.31 € |
| Jefe del proyecto | 1.637,50 € | 540,37 € | 3 | 25% | 1.606,41 € |
| Coste Total | | | | | 8.399,53 € |

Tabla Desglose de costes de personal

Una vez vistos todos los costes iniciales, es necesario ver los costes indirectos asociados al proyecto. Se consideran costes indirectos los gastos extra asociados a cada proyecto necesarios para su realización. En estos se incluyen los gastos de funcionamiento de la empresa (luz, agua, conexión a internet, alquiler, etc.) y los posibles gastos imprevistos. La empresa establece un 20% al coste inicial del proyecto.

Teniendo los gastos anteriormente descritos, el coste total del proyecto se puede resumir en la siguiente tabla :

|  |  |
| --- | --- |
| Recurso | Coste |
| Recursos materiales | 14,29 € |
| Recursos software | 101,18 € |
| Recursos hardware | 1.000 € |
| Recursos humanos | 8.399,53 € |
| Costes indirectos (20%) | 1.903,00 € |
| Coste Total | 11.418,00 € |

Tabla Desglose de costes iniciales del proyecto.

# Bibliografía

|  |  |
| --- | --- |
| [1] | R. S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 2014. |
| [2] | K. S. &. J. Sutherland, «www.scrumguides.org,» Noviembre 2017. [En línea]. Available: https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100. |
| [3] | Amortización de inmovilizado, «www.plangeneralcontable.com,» [En línea]. Available: https://www.plangeneralcontable.com/?tit=amortizacion-de-inmovilizado-metodo-lineal-o-de-cuotas-fijas&name=Manuales&fid=el0bcac. |
| [4] | MINISTERIO DE EMPLEO Y SEGURIDAD, «https://www.boe.es,» [En línea]. Available: https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf. |

1. Nombre de la aplicación que se va a desarrollar [↑](#footnote-ref-1)