**Proyecto: Complutensers**

**Plan del proyecto del software**

**(Formato Resumido Pressman)**

**Miembros del equipo:**

|  |  |
| --- | --- |
| Bittor Alaña Olivares | Carlos Moreno Morera |
| Claudia Guerrero García-Heras | Jorge Sainero Valle |
| Manuel Suárez Román | Rubén Ruperto Díaz |
|  |  |
|  |  |

**Control de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de versión** | **Fecha** | **Autores** | **Descripción** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Índice

1. Introducción

1.1 Propósito del plan

1.2 Ámbito del proyecto y objetivos

1.2.1 Declaración del ámbito

1.2.2 Funciones principales

1.2.3 Aspectos de rendimiento

1.2.4 Restricciones y técnicas de gestión

1.3 Modelo de proceso

2. Estimaciones del proyecto

2.1 Datos históricos

2.2 Técnicas de estimación

2.3 Estimaciones de esfuerzo, coste y duración

3. Estrategia de gestión del riesgo

3.1 Introducción: Estudio de los riesgos

3.2 Priorización de riesgos del proyecto

3.3 Plan de gestión del riesgo Reducción, supervisión y gestión del riesgo

3.4 Planificación temporal del Control de Riesgos

3.5 Resumen

4. Planificación temporal

4.1 Estructura de descomposición del trabajo/Planificación temporal

4.2 Gráfico Gantt

4.3 Red de tareas

4.4 Tabla de uso de recursos

5. Recursos del proyecto

5.1 Personal

5.2 Hardware

5.3 Software

6. Organización del personal (Gestión del Equipo)

6.1 Estructura de equipo

6.2 Informes de gestión

7. Mecanismos de seguimiento y control

7.1 Garantía de calidad y control (Plan de Calidad)

7.2 Gestión y control de cambios (Plan GCS)

7.2.1 Introducción: Propósito, Alcance, Definiciones, Referencias

7.2.2 Tipos de artefactos a gestionar (los ECSs)

7.2.3 Criterios y protocolos para Nombrar los ECSs

7.2.4 Responsable de los procedimientos de GCS y de la creación de Líneas Base.

7.2.5 Políticas para el Control de Cambios y la Gestión de Versiones

7.2.6 Registros para mantener el rastro de los cambios

8. Apéndices

1. **Introducción**
   1. **Propósito del plan**

Con los más de 84.000 estudiantes con los que cuenta la UCM es muy difícil hacer llegar toda la información a todos ellos. Además, la mayor parte de ellos ven la Universidad como un lugar por el que tienen que pasar para llegar a alcanzar un futuro laboral.

La carencia del sentimiento de pertenencia a una entidad como lo es la Universidad Complutense de Madrid, repercute en la participación y organización de actividades extracurriculares. Con el propósito de solventar el problema de desinformación y falta de participación, pretendemos con esta aplicación unificar en un solo lugar toda la información sobre agrupaciones, mapas de facultades, actividades y eventos de la Universidad Complutense, para uso de toda persona vinculada con la UCM. Para ello, adaptaremos una página web que contiene ya toda esta información, a una aplicación móvil más manejable y práctica gracias a la cual la gente podrá acceder a ella con mayor facilidad.

* 1. **Ámbito del proyecto y objetivos**
     1. **Declaración del ámbito**

Con la aplicación “Complutensers” queremos llegar a todas aquellas personas vinculadas de alguna manera con la UCM y que tenga interés en algún ámbito diferente del de los estudios que en la Universidad se imparten.

* + 1. **Funciones principales**

Las funciones principales de la aplicación serán:

* Búsqueda de lugares: cualquier persona relacionada con la UCM podrá buscar cualquier lugar de la UCM en el mapa de la aplicación.
* Gestión de actividades: todas las Agrupaciones de la UCM podrán proponer nuevas actividades para llevar a cabo y colgarlas en la app para que cualquiera que disponga de ella acceda a la información, además de pedir colaboración a los usuarios.
* Apuntarse a actividades: todo usuario registrado podrá apuntarse para participar en las actividades que se desarrollen en la UCM.
* Búsqueda de actividades: los usuarios podrán buscar actividades acorde a sus intereses.
  + 1. **Aspectos de rendimiento**

No hemos encontrado problemas de rendimiento, ya que todo lo relacionado con almacenar datos será gestionado a través de la Base de Datos utilizada para la página web http://eventos.ucm.es.

* + 1. **Restricciones y técnicas de gestión**

El proyecto carece de restricciones técnicas ya que todo lo referido al mantenimiento del hardware y del software es responsabilidad de la Universidad Complutense de Madrid.

Técnicas de gestión: Un modelo intermedio entre Descentralizado Democrático y Descentralizado Controlado

* 1. **Modelo de proceso**

Como modelo de proceso hemos elegido el modelo de proceso unificado con unos pequeños rasgos del modelo en espiral (ya que tenemos intención de realizar el proyecto de forma incremental) y también con unos pocos rasgos de Scrum (reuniones periódicas y documentación diaria).

1. **Estimaciones del proyecto**
   1. **Datos históricos**

La UCM llevó a cabo un proyecto similar, en el que se puso en marcha una página web para ofrecer información sobre las actividades que tienen lugar en la universidad. Este proyecto fue un fracaso, resultando en una página web sin visitas, y que los estudiantes ni siquiera conocen.

Nuestro objetivo es recuperar esta idea en una aplicación móvil, mucho más atractiva y práctica para los estudiantes, de forma que este servicio tenga más usuarios y la comunidad universitaria participe más activamente en los eventos propuestos.

Al ser el primer contacto de nuestro grupo con las técnicas de ingeniería del software, no tenemos la experiencia necesaria para hacer una estimación muy precisa.

Estimamos que los subsistemas de gestión de usuario y actividades serán comparables, cada uno, a una práctica de programación de primero de un mes de duración, es decir, 4 sesiones de 2 horas en clase, más otras 2 sesiones de dos horas para completar. Habría que añadir un periodo de planificación anterior, que no había en las prácticas ya que la planificación estaba hecha en el enunciado.

El subsistema buscador es más pequeño, y en consecuencia necesitaría menos horas.

* 1. **Técnicas de estimación**

Aún no tenemos los conocimientos suficientes en métricas como para realizar una estimación exhaustiva de los costes del proyecto, con lo cual, nos limitamos a realizar la estimación del esfuerzo y coste en horas de trabajo.

* 1. **Estimaciones de esfuerzo, coste y duración**

En primer lugar, estimaremos el tiempo que habrá que dedicar a cada subsistema para completarlo, basándonos en otras experiencias de programación en la carrera de informática. Vamos a diferenciar 4 fases:

* Análisis: Incluye el análisis de los casos de uso, la búsqueda de información para resolver el problema y la planificación de las funciones, clases, etc. del programa.
* Diseño: Incluye el diseño gráfico y la programación de la interfaz.
* Codificación: Incluye el desarrollo del código propiamente dicho.
* Pruebas: Incluye las pruebas del código y corrección de errores.
* Identificamos 3 subsistemas: Buscador, Gestor de Usuarios y Gestor de Actividades.

Buscador:

* Diseño: 4 horas
* Análisis: 3 horas
* Codificación: 4 horas
* Pruebas: 3 horas

Gestor de Usuarios:

* Diseño: 8 horas
* Análisis: 5 horas
* Codificación: 10 horas
* Pruebas: 4 horas

Gestor de Actividades:

* Diseño: 8 horas
* Análisis: 5 horas
* Codificación: 14 horas
* Pruebas: 4 horas

Obtenemos un total de 72 horas de trabajo con estas estimaciones. Entre los 6 miembros del equipo, cada uno tendría una carga de 12 horas. Si suponemos que cada miembro puede dedicar 2 horas a la semana, la duración aproximada para completar los subsistemas serían 6 semanas.

A esto hay que añadir las horas dedicadas a redactar informes y las reuniones de los miembros del equipo.

1. **Estrategia de gestión del riesgo**
   1. **Introducción: Estudio de los riesgos**

Será primordial evitar todo agravante. Como cualquier aplicación móvil o idea de start-up, el riesgo es inherente al proyecto y las probabilidades de éxito son más bien bajas, por lo que sólo mediante un trabajo fino y una apropiada estrategia de prevención y actuación para riesgos se conseguirá una salida adecuada para el producto. Es muy habitual en una aplicación que se lanza a un público muy amplio que la acogida no sea demasiado amplia.

También puede haber problemas con el personal que desarrolla el proyecto, bien sea por no estar plenamente capacitados, por retiradas de miembros importantes del equipo… Además, el proyecto *Complutensers* depende fuertemente de sistemas externos, por tanto es imperativo para su desarrollo planeado que se obtengan todos los permisos, autorizaciones y colaboraciones pertinentes. Su ausencia supondría un riesgo significativo.

* 1. **Priorización de riesgos del proyecto**

Primer riesgo: denegación de permisos o colaboración. Sin la ayuda de las instituciones complutensinas se dificultaría increíblemente el trabajo, ya que el acceso al Campus Virtual y a la información relativa a las actividades y grupos son uno de los puntos más fuertes de la aplicación.

Segundo riesgo: retirada o incapacidad del equipo. El proyecto ha de tener un personal competente y atento. Como está planeado que lo lleven adelante estudiantes, no debería haber problemas de transfuguismo a otras compañías incentivado por más alta remuneración, no obstante puede ser que los estudiantes no tengan la experiencia necesaria para un proyecto de este calibre, o que por exámenes u otras razones se caigan del equipo. Podría hacer tambalear el proyecto. Se necesitará gran constancia por parte de los integrantes del grupo, no sólo a la hora de desarrollar la aplicación sino después, para incentivar su uso y mantenerla activa.

Tercer riesgo: acogida pobre. Claramente la aplicación se hace para que se utilice. Una acogida pobre sería un golpe duro a la motivación del equipo. Por fortuna no supone un problema por presupuesto, no hay que pagar salarios a estudiantes y el uso de la aplicación es totalmente gratuito por lo que mayor acogida no se traduciría en más dinero, pero sí que haría que las instituciones de la universidad pusieran más medios a disposición de los jóvenes. Es por ello que una acogida baja puede bajar la moral de los desarrolladores y dejar el proyecto pendiendo de un hilo, ya que su constancia requerirá de incentivación (sobretodo moral) por parte del público y de la UCM

* 1. **Plan de gestión del riesgo Reducción, supervisión y gestión del riesgo**

**3.1. Denegación de permisos o colaboración**

3.1.1. Reducción: se trabajará intensamente la comunicación con los profesores de la Facultad de Informática, los cuales podrán hacer de puente para conseguir el beneplácito de los organismos de dirección, los cuales podrán facilitar la colaboración de la universidad. Para las plataformas de Google que se vayan a utilizar, se procederá a estudiarlas profundamente mucho antes de empezar a programar.

3.1.2. Supervisión: este riesgo se concentra mayoritariamente en el comienzo del proyecto, por lo que su supervisión consistirá básicamente en mantener una estrecha colaboración con todos los agentes implicados y estar bien informados estudiando todos los permisos a fondo durante las primeras etapas de desarrollo. Posteriormente se hará un esfuerzo por mantener estos buenos lazos, en vista de posibles extensiones o adición de propiedades.

3.1.3. Plan de Contingencia: si el riesgo se materializa, tras un debate sobre la viabilidad del proyecto, se buscarán plataformas alternativas y métodos de obtener la información que estas colaboraciones nos aportarían de otras maneras. Básicamente se tratará de un proceso de mentalización de que, a pesar de añadir una ingente carga de trabajo, la viabilidad sería aún posible.

**3.2. Retirada o incapacidad del equipo**

3.1.1. Reducción: se insistirá en crear sentimiento de grupo, cierta camaradería e implicación para que nadie se retire del proyecto. Para paliar la posible inexperiencia, se estrecharán los lazos de colaboración entre aquellos que más experimentados sean con los que menos. Será muy importante que quienes menos capacidad tengan sean conscientes de ello y se pongan en completa disposición de sus mentores, quienes también tendrán que hacer un gran esfuerzo en no ensimismarse con su parte del proyecto sin hacer caso de los menos expertos. Por tanto la reducción de este riesgo consiste en tener un grupo unido, en el que uno aprende de otro y se espera total compromiso.

3.1.2. Supervisión: la supervisión de este riesgo consistirá en analizar las relaciones del grupo, en conocer (sin ser invasivos) los compromisos externos de cada integrante del equipo y en evaluar las relaciones personales entre los mismos.

3.1.3. Plan de Contingencia: en caso de que el riesgo se dé, el procedimiento a seguir será el de sustitución de personal, buscando en la Facultad de Informática a personas capacitadas e implicadas. Se extenderá la palabra entre los círculos de estudiantes, se consultará la opinión de los desarrolladores sobre sus compañeros de clase y se atenderán sus recomendaciones.

**3.3. Acogida pobre**

3.1.1. Reducción: la reducción de este riesgo se basa casi exclusivamente en la promoción. Dar a conocer el producto será fundamental para que el proyecto funcione adecuadamente. La gente deberá saber del producto incluso antes de su lanzamiento, se procurará crear expectación alrededor de él. Se realizará una gran labor informativa, tanto por las facultades como por las calles de la Ciudad Universitaria. Además se trabajará con las agrupaciones, a las cuales se les intentará hacer entender el gran rédito que podrán sacar del producto, para que sean los primeros en incentivar su uso entre aquellos estudiantes inscritos en dichas agrupaciones. Si el producto aspira a incentivar la vida de ocio, cultura y recreación de la universidad, será primordial que dicho mundo esté incorporado al funcionamiento de la aplicación desde el primerísimo instante.

3.1.2. Supervisión: la supervisión de este riesgo deberá hacerse con gran periodicidad. Habrá gente encargada de monitorear cómo se acoge la aplicación. Al ser un riesgo muy gradual, se necesitará constancia en la supervisión. Se analizarán datos de descargas, se mantendrán charlas con directores de agrupaciones, y se pondrá especial énfasis en atender todas las quejas, reclamaciones y sugerencias que los usuarios hagan para que la aplicación se adapte a lo que busca el público al que va dirigida*.*

3.1.3. Plan de Contingencia: si se advierte una baja actividad, se incrementará la labor promocional, se pondrán más carteles, se hablará con las agrupaciones, se intentarán meter anuncios en las páginas de las Facultades...

* 1. **Planificación temporal del Control de Riesgos**

Como hemos comentado, hay tres riesgos prioritarios, pero cada uno atiende a tiempos distintos. En las etapas más tempranas del proyecto, se realizará una evaluación periódica de las relaciones del grupo y las relaciones con las instituciones. En varias reuniones semanales se debatirá cómo marchan los permisos necesarios y la química del equipo. También se irá introduciendo el debate sobre la promoción.

Una vez se asegure que las relaciones entre los miembros y la universidad van por buen camino, se podrá insistir algo más en la promoción del producto, no obstante en aquellos momentos en los que la escritura del código haya empezado se empezará a insistir en dar a conocer el proyecto, si bien no a todo el público al que va dirigido, sí al personal de profesores y todos aquellos que en la UCM estén envueltos en la dirección/administración de actividades de ocio, cultura, arte y recreación.

Pasado este momento, se realizará una reunión semanal de evaluación de la situación. Se discutirán las relaciones del grupo y el conocimiento de la aplicación entre el público. Se deberá tener aún una actitud cauta respecto al riesgo de la falta de colaboración y permisos, ya que aunque atiende a un problema concentrado más en el comienzo del proyecto, mantener buenas relaciones con la universidad podrá ser muy útil en el futuro.

* 1. **Resumen**

Siendo *Complutensers* una aplicación móvil dirigida a un público bastante amplio, puede ser que no tenga buena acogida. La dependencia que tiene en otras plataformas pueden vulnerar su recorrido, y la calidad de estudiantes de sus desarrolladores puede poner en riesgo el proyecto. Por tanto, será de suma importancia lubricar las relaciones con las instituciones, promocionar el producto entre las agrupaciones y crear estrechos vínculos entre los miembros del grupo, además de entre los directores del proyecto y las agrupaciones y las instituciones de la Universidad Complutense de Madrid.

1. **Planificación temporal**
   1. **Estructura de descomposición del trabajo/Planificación temporal**

La realización del trabajo viene fuertemente marcada por las sucesivas entregas que, a lo largo del curso, tenemos que ir realizando. En primer lugar, antes de empezar a realizar la propia aplicación, precisábamos de la documentación necesaria (plan de proyecto, plan SRS…) y por ello puede decirse que trabajamos en cascada. Esto es así porque de otra manera no podríamos determinar que pantallas, que métodos… iba a utilizar nuestra aplicación. Para realizar dicha documentación nos dividimos las tareas. Para los CU nos dividimos en tres equipos, uno para cada gestor (usuarios, actividades y búsqueda). Después para realizar las especificaciones y el plan de proyecto temporal repartimos los apartados de manera nominal.

La forma en la que organizaremos el proyecto será la siguiente:

* **Documentación:** realización de los primeros documentos en los que se describa a grandes rasgos el proyecto, el equipo y una breve descripción de los requisitos y funcionamiento interno de la aplicación
* **Diseño:** se realizará el diseño de las diferentes interfaces y pantallas de la aplicación dependiendo, estas pantallas vendrán asociadas a sus respectivos CU´s, así como la determinación de un estilo común para que todas las pantallas queden integradas
* **Recopilación:** se generará un documento SRS, en el cual vendrán especificados meticulosamente todos los puntos de la aplicación. Es importante realizar un SRS completo puesto que será nuestra base para las siguientes fases del proyecto.
* **Realización:** por última parte, llevaremos la aplicación (o alguna de sus partes) a la vida real y escribiremos el código de la misma y comprobaremos que el resultado obtenido se corresponde con lo planificado en los pasos previos del proyecto.
  1. **Gráfico Gantt**
  2. **Red de tareas**
  3. **Tabla de uso de recursos**

1. **Recursos del proyecto**
   1. **Personal**

El grupo Ariadna consta de seis miembros, repartidos en función al subsistema de la aplicación al que se dedican:

* 2 personas en el subsistema de gestión de actividades. Una de ellas dedicada a revisar los casos de uso y la otra encargada del diseño del programa.
* 3 personas en el subsistema de gestión de usuarios. La estructura es similar a la del subsistema anterior pero como el número de casos de uso es mayor tenemos dos revisores.
* 1 persona en el subsistema del buscador. Esta persona se encarga de revisar los casos de uso y de diseñar el programa.
* 2 diseñadores de la aplicación que se encargarán de diseñar la parte gráfica de la aplicación.
* 1 experto en plataformas móviles para ayudar a los programadores de la aplicación.
* 1 gestor de programación encargado de coordinar a los programadores y a los diseñadores del programa.
* 5 programadores.

Varias personas ejercen varias funciones.

* 1. **Hardware**

5.2.1 Hardware de desarrollo

El hardware empleado para el desarrollo consta de los ordenadores de los programadores, un dispositivo con Android, otro con iOS.

Los ordenadores de los programadores son ordenadores comunes, no se necesitarán grandes especificaciones para desarrollar la aplicación.

5.2.2 Hardware del cliente

El cliente necesitará un dispositivo con conexión a internet vía wifi.

* 1. **Software**

5.3.1 Software de desarrollo

* Eclipse: Entorno que usarán los programadores que permite el uso de Java.
* Google Drive: Para compartir los archivos entre los integrantes del equipo.
* Microsoft office: Para la documentación.
* AppMaker: Para el diseño de la aplicación móvil.

5.3.2 Software del cliente

El cliente necesitará que su dispositivo tenga una versión de Android superior o igual a la 4.0.1 o una versión de iOS superior o igual a la 6.0.0.

1. **Organización del personal (Gestión del Equipo)**
   1. **Estructura de equipo**

La organización del equipo mezcla características del modelo descentralizado democrático (DD) y descentralizado controlado (DC).

Por un lado, la comunicación entre los miembros del equipo es horizontal, se debaten propuestas y se plantean dudas de forma conjunta y se toman decisiones por acuerdo de todos los miembros del grupo. Esto tiene la ventaja de que la comunicación es más ágil y hay una mayor riqueza de ideas.

Por otro lado, hemos visto que es necesario tener un jefe de equipo, encargado de supervisar y repartir el trabajo entre los miembros. Hemos nombrado a Carlos Moreno para esta función, lo que nos permite ponernos a trabajar de forma más organizada y nos ahorra tiempo a la hora de asignar las tareas.

Los 6 miembros trabajamos como programadores y diseñadores, y nos agrupamos en grupos más reducidos para centrarnos en cada subsistema. Estos subgrupos trabajan de forma democrática, sin un subjefe.

* 1. **Informes de gestión**

El reparto de miembros del equipo en cada subsistema es el siguiente:

Subsistema gestor de actividades: Claudia Guerrero y Carlos Moreno

Subsistema gestor de usuarios: Bittor Alaña, Jorge Sainero Valle y Manuel Suárez

Subsistema buscador: Rubén Ruperto

1. **Mecanismos de seguimiento y control**

* 1. **Garantía de calidad y control (Plan de Calidad)**

En un proyecto como este, en el que ya hay muchas aplicaciones parecidas en el mercado intentando realizar el mismo propósito de forma fallida, es muy importante tener un buen plan de calidad. Hacer una buena aplicación que marque la diferencia determinará el éxito de nuestro proyecto. Lo primero que nos planteamos fue porque otras aplicaciones con la misma idea de fondo fallaron; encontrar importante no solo solucionar sus fallos sino además ofrecer nuevas funcionalidades, hacerlo más práctico, más personalizado, más útil y más llamativo para el público.

Para ello para ello hacemos reuniones periódicas en las que debatimos continuamente tanto sobre el trabajo realizado, sobre el que queda por hacer. El *brainstorming* es una de las herramientas que usamos para ello. Sobre el ya realizado para comprobar que el producto final es lo que esperábamos y sobre el que queda por hacer para ver en qué podemos innovar, en que podemos marcar la diferencia.

Aunque en última instancia el jefe de equipo se encarga de que el trabajo realizado por cada uno cumpla los estándares que se exigían, cada miembro del grupo es responsable de realizar el trabajo conforme sea necesario, ajustándose a los requisitos y haciendo un trabajo de calidad.

* 1. **Gestión y control de cambios (Plan GCS)**
     1. **Introducción**

Una de las tareas más difíciles a las que nos hemos enfrentado sin duda hasta el momento es la coordinación del trabajo. En un proyecto de esta envergadura, en el que trabajamos simultáneamente seis persona es necesario saber no solo quién hace qué y cuándo lo hace, sino poder tener acceso a los detalles menores de su trabajo, ya que en ocasiones necesitamos conocerlo para poder continuar con nuestra asignación.

Para ello consideramos indispensable la comunicación, sin embargo realizar reuniones periódicas para hablar sobre el proyecto es algo que hacemos cuando el jefe de grupo lo considera necesario ya que en la mayoría ocasiones la comunicación es vía online, ya que nos parece un método muy práctico, por tanto sería útil una herramienta

Para ello necesitamos una herramienta que nos permita:

* Almacenar todo el trabajo hecho hasta la fecha.
* Acceder a versiones anteriores del proyecto.
* Trabajar simultáneamente en el mismo proyecto.
* Saber qué cambios se han hecho en el trabajo.
* Comunicarnos, es decir, comentar el trabajo de otros colaboradores.
  + 1. **Tipos de artefactos a gestionar (los ECSs)**

Hasta ahora en el proyecto sólo hemos necesitado compartir documentos, imágenes y archivos. Es decir aún no hemos empezado a compartir código.

* + 1. **Criterios y protocolos para Nombrar los ECSs**

Tenemos todo el trabajo en una única ubicación organizado en carpetas por fecha de entrega en primer lugar, ya que tener el trabajo priorizado es una actividad muy importante para la coordinación del equipo. Estas vienen marcadas tanto por las exigencias externas al proyecto, como puede ser el cliente, como por el jefe de equipo.

Dentro de cada fecha, el trabajo está organizado por tareas y temática, de modo que para encontrar un archivo es más importante que sepas qué estás buscando en lugar del nombre. A su vez cada archivo tiene como nombre el nombre de la parte del proyecto a la que pertenecen seguido de una barra baja y un par de palabras que describan lo que contiene. Por ejemplo: *Entrega 4 >> Pantallas >> Pantalla\_logout\_estudianteUCM.*

* + 1. **Responsable de los procedimientos de GCS y de la creación de Líneas Base.**

Aunque contamos con la figura del jefe de grupo para que coordine el trabajo, en nuestro equipo el trabajo de coordinación se procura realizar de forma horizontal, cada uno se encarga de que el trabajo realizado sea correcto y esté coordinado con el resto en la medida de lo posible. Ante cualquier duda se consulta al resto del grupo para la toma de decisiones importantes.

Antes de dar cualquier trabajo por finalizado el jefe de grupo se encargará o bien de revisarlo o bien de encargar a una única persona que revise el trabajo en su conjunto.

* + 1. **Políticas para el Control de Cambios y la Gestión de Versiones**

Para esta etapa del proyecto nos ha parecido que la mejor herramienta era Google Drive ya que nos permitirá compartir docs, pdfs, imágenes, diapositivas, tablas... Nos ha parecido idónea debido a que no solo podemos trabajar a la vez en un mismo documento viendo al momento el trabajo del resto y facilitando la coordinación del trabajo para que éste tenga cohesión, sino que también notifica de los cambios hechos en el documento desde la última vez que lo visualizaste indicándote que se ha cambiado y quién lo ha hecho. Es la herramienta que usamos tanto para trabajar como para el control de versiones.

Además esta plataforma de trabajo nos permite trabajar con nuestro correo institucional lo cual es bastante cómodo.

El proceso general de realización de un trabajo es:

1. Análisis de la envergadura del trabajo por parte del jefe que hace un esquema sobre cómo dividirlo y quién es la persona más adecuada para cada parte del proyecto.
2. El resto del equipo da el visto bueno al reparto.
3. Se empieza a trabajar sobre una plantilla, modificando cada uno su parte y dejando comentarios en las partes de los otros colaboradores.
4. Se puede comentar tu propio trabajo para consultar sobre él.
5. Se revisa el trabajo final.

Cuando tengamos que empezar a trabajar con código nos plantearemos si merece la pena pasarse a una plataforma con control de versiones en el código, como podría ser GitHub.

* + 1. **Registros para mantener el rastro de los cambios**

La herramienta que usamos guarda automáticamente los cambios realizados, no obstante guardamos en un repositorio las diferentes versiones de las partes vitales del proyecto.

1. **Apéndices**