Laura Marco

María Juberías

Alejandro Ramírez

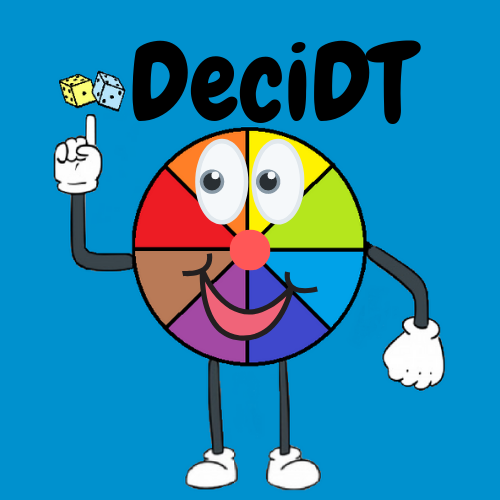
Eduardo López

Daniel Jiménez

Mario Jiménez

Miguel Macicior

María Juberías



**DECIDT**

**DECIDT**

PLAN DE PROYECTO

ÍNDICE

[1. INTRODUCCIÓN 2](#_Toc1217278)

[1.1 ¿QUÉ ES DeciDT? 2](#_Toc1217279)

[1.2 ÁMBITO DEL PROYECTO 3](#_Toc1217280)

[1.3 FUNCIONES PRINCIPALES 4](#_Toc1217281)

[1.4 ASPECTOS DE RENDIMIENTO 6](#_Toc1217282)

[1.5 MODELO DE PROCESO Y TÉCNICAS DE GESTIÓN 6](#_Toc1217283)

[2. ESTIMACIONES DEL PROYECTO 7](#_Toc1217284)

[2.1 INTRODUCCIÓN 7](#_Toc1217285)

[2.2 DATOS HISTÓRICOS 7](#_Toc1217286)

[2.3 PLAN DE PROYECTO 8](#_Toc1217287)

[3. ESTRATEGIA DE GESTIÓN DEL RIESGO 11](#_Toc1217288)

[3.1 IDENTIFICACIÓN DE RIESGOS DEL PROYECTO 12](#_Toc1217289)

[3.2 PRIORIZACIÓN DE RIESGOS DEL PROYECTO 14](#_Toc1217290)

[3.3 PLAN DE GESTIÓN DE RIESGO 16](#_Toc1217291)

[3.4 PLANIFICACIÓN TEMPORAL DEL CONTROL DE RIESGO 20](#_Toc1217292)

[4. PLANIFICACIÓN TEMPORAL 21](#_Toc1217293)

[4.1 ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO 21](#_Toc1217294)

[4.2 DIAGRAMA DE GANTT 27](#_Toc1217295)

[4.3 GESTIÓN DE RECURSOS 27](#_Toc1217296)

[5. RECURSOS DEL PROYECTO 28](#_Toc1217297)

[5.1 PERSONAL 28](#_Toc1217298)

[5.2 HARDWARE Y SOFTWARE 28](#_Toc1217299)

[**HARDWARE** 28](#_Toc1217300)

[**SOFTWARE** 29](#_Toc1217301)

[6. Plan de GCS / Mecanismos de Gestión y Control 31](#_Toc1217302)

[6.1 Garantía de calidad y control 31](#_Toc1217303)

[**Conceptos Clave** 31](#_Toc1217304)

[6.2 Gestión de la configuración 32](#_Toc1217305)

[6.3 Control de Versiones 34](#_Toc1217306)

[6.4 Líneas Base 34](#_Toc1217307)

[7. GESTIÓN DE EQUIPO 36](#_Toc1217308)

[7.1. Estructura del equipo 37](#_Toc1217309)

[7.2 Informes de gestión 38](#_Toc1217310)

[8. ANEXOS Y REFERENCIAS 40](#_Toc1217311)

[**ANEXO** 40](#_Toc1217312)

[**REFERENCIAS** 41](#_Toc1217313)

# 1. INTRODUCCIÓN

**Autor:** Laura Marco

**Supervisa:** María Juberías

## 1.1 ¿QUÉ ES DeciDT?

Cualquier decisión significa trazar un camino y despedirse de otro.

Elegir y descartar.

Tomar y dejar.

Y cuando esta decisión es de las difíciles, de las que pueden marcar un antes y un después, esta doble dimensión se nos hace más presente, más visible. Se nos graban a fuego esos momentos en los que estamos en la encrucijada y nuestra mente puede volver a ellos con mucha facilidad. Cuanto más difícil, más intenso. Cuanto más complicada nos resulta la elección, más son las posibilidades de que nos surja la duda.

Es ahí donde surge DeciDT, cuando nos encontramos con la incapacidad de **tomar una decisión** debido a la amplia variedad de opciones en el entorno en el que vivimos.

Es una plataforma que pretende **solucionar este problema**, y ayudar a los usuarios a solucionar sus dudas sobre cualquier tema, desde a qué restaurante ir a cenar hasta si se debe aceptar una oferta laboral o no. El rango de decisiones es muy amplio y DeciDT es un proyecto enfocado a solucionar cualquier duda que te pueda surgir en cualquier momento o sobre cualquier tema. DeciDT nace como una aplicación dirigida a todos los públicos con el fin de resolver cualquier duda que nos pueda surgir a lo largo de nuestra vida.

Nuestra aplicación es una herramienta que se presenta para solucionar los distintos problemas:

* Poder hacer decisiones aleatorias con unos mecanismos en función de las opciones a elegir, desde una moneda a cara o cruz, hasta una ruleta con varias soluciones.
* La posibilidad de lanzar una pregunta a la gente que forma parte de la plataforma, cuando la decisión es mas seria. Si esa misma pregunta ha sido lanzada por un usuario anteriormente podremos ver las respuestas dadas anteriormente por otros usuarios.
* La posibilidad de hacer y gestionar torneos con otros usuarios.

## 1.2 ÁMBITO DEL PROYECTO

DeciDT es una **herramienta** que nos ayudará a tomar decisiones tanto rápidas e insignificantes como aquellas importantes para nuestro futuro y servirá para elegir un restaurante entre 4 tanto como para estructurar torneos y equipos o para realizar una votación o encuesta.

Para ello hemos desarrollado una plataforma que nos ayude a solucionar las distintas dudas que le puedan surgir a las personas, y en función del tipo de pregunta que hagan proporcionarles una respuesta u otra.

Para llevarlo a cabo hemos dividido nuestro sistema en 3 subsistemas:

* Una sección dedicada a la **gestión de usuarios**, que consiste en añadir o eliminar usuarios, y decidir de qué forma quieren participar si para realizar una pregunta o para responder.
* Otra segunda sección está dedicada a la **gestión de votaciones**, en la que nos centramos en almacenar las votaciones de los distintos usuarios y proporcionárselas al usuario que realizó la pregunta. Además gestionamos un almacenamiento de estadísticas y preguntas anteriores que puedan servir de ayuda.
* Y la última sección está dedicada a la **gestión de algoritmos de aleatoriedad** y está compuesta por las funciones que generan estos algoritmos para solucionar aquellas preguntas cuya respuesta son inmediatas y aleatorias.

De esta manera nuestro sistema queda dividido en 3 subsistemas,

## 1.3 FUNCIONES PRINCIPALES

|  |  |  |
| --- | --- | --- |
| **GENERAL** | Iniciar sesión / Cerrar sesión. | Permite a los usuarios abrir o cerrar su sesión |

|  |  |  |
| --- | --- | --- |
| **SUBSISTEMA DE GESTIÓN DE USUARIOS** | Registrar Usuario | Permite registrarse a los usuarios que no lo están. |
| Organizar torneo | Permite a los usuarios registrados organizar torneos |
| Realizar Votación | Permite a los usuarios votar |
| Decisión aleatoria | Permite a los usuarios realizar una decisión aleatoria. |
| Agregar amigos | Permite a los usuarios registrados agregar amigos. |

|  |  |  |
| --- | --- | --- |
| **SUBSISTEMA DE GESTIÓN DE VOTACIONES** | Registrar las votaciones | El sistema registra las votaciones de cada usuario y las contabiliza. |
| Almacenamiento de las votaciones | El sistema almacena las votaciones y las preguntas. |
| Dar feedback del resultado | Una vez acabado el tiempo, el sistema proporciona las respuestas. |
| **SUBSISTEMA DE ALGORITMOS ALEATORIOS** | Generación de algoritmos aleatorios. | El sistema genera un algoritmo aleatorio, en forma de moneda, ruleta o dado. |

## 1.4 ASPECTOS DE RENDIMIENTO

Consideramos la duración media de la sesión como unos 10 minutos por cada vez que se abra la aplicación. Y teniendo en cuenta que la presencia en el mercado de aplicaciones de este tipo es casi inexistente, y que en España el número de descargas por día es de 4 millones [[1]](#_REFERENCIAS) y que lo que más valoran los españoles en una app es que cubra una necesidad (72%) entonces

Podemos deducir que como nuestra app está enfocada a solucionar una necesidad que pocas aplicaciones resuelven, las descargas serán notables y el grado de adicción a ella irá incrementando según se vaya usando.

## 1.5 MODELO DE PROCESO Y TÉCNICAS DE GESTIÓN

Para la realización y desarrollo del proyecto hemos usado el Proceso Unificado de Desarrollo [[2]](#Pud) , que está dirigido por los casos de uso, centrado en la arquitectura y centrado en identificar los posibles riesgos.

# 2. ESTIMACIONES DEL PROYECTO

**Autor:** Mario Jiménez Gutiérrez

**Supervisa**: Miguel Macicior

## 2.1 INTRODUCCIÓN

Exponemos una estimación de costes los costes temporales y financieros del proyecto. Es importante recalcar ciertos aspectos: no existe una gran dificultad en programarlo, aunque a la hora de optimizarlo, se demandarán algoritmos de búsqueda y patrones; no obstante, en dicha versión del modelo, no se utilizarán estos dado que los integrantes del grupo todavía no poseen esos conocimientos. Por otro lado, el acceso a la base de datos tampoco supondrá gran complejidad.

Con el objeto de calcular el esfuerzo y el coste del proyecto utilizaremos la técnica de **Descomposición Basada en el Proceso.**

## 2.2 DATOS HISTÓRICOS

Recalcando que no se necesita un conocimiento extenso de algoritmia y de lenguajes de programación para el desarrollo del proyecto estimamos que el coste del proyecto es de **450LDC/PM**. De esta forma, y basándonos en los salarios actuales de ingenieros informáticos junior sin apenas experiencia en el mercado laboral, hemos decidido establecer unas nóminas de **2400€/PM** a nuestros trabajadores; cotizando además **400€/PM a la Seguridad Social**, lo que suma un **total de 2800€/PM**.

Como somos 7 integrantes, el coste mensual sería de:

2800∗7=19600€/𝑚𝑒𝑠.

Teniendo en cuenta que 16800**€ son salarios íntegros** y que **2800€ son cotizaciones** para la Seguridad Social.

Por otro lado, la aplicación **necesita una base de datos** donde poder almacenar tanto los datos de usuarios como preguntas y encuestas. Para realizar las pruebas de la versión beta, será necesario **contratar varios servidores PHP** que conectan con la base de datos, cuyo **precio es de 150€ al mes**.

Estimamos la necesidad de **2 servidores**, lo que nos llevaría a un gasto de: 150∗2=300 €/𝑚𝑒𝑠.

Por tanto, tendríamos un **gasto final** de: 19900 €/𝒎𝒆𝒔

## 2.3 PLAN DE PROYECTO

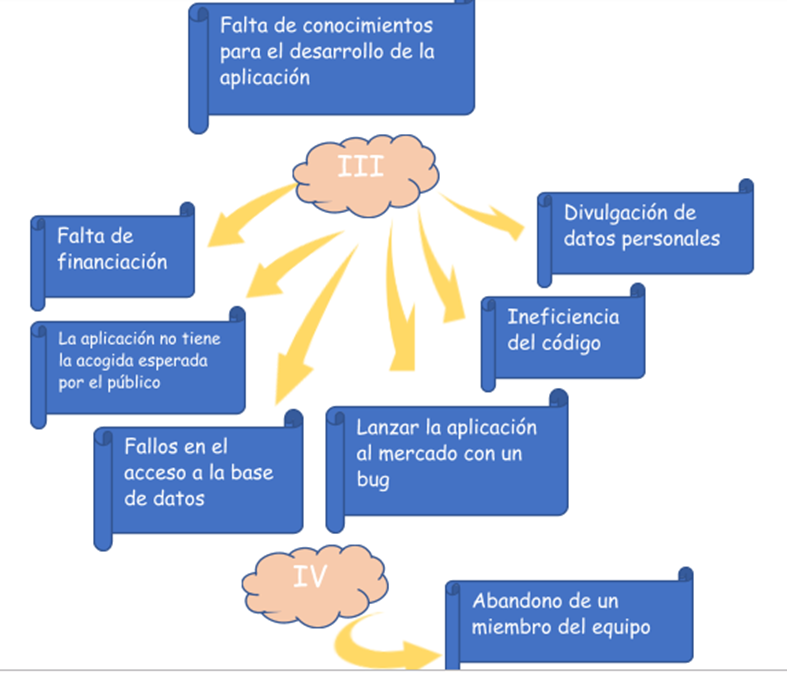
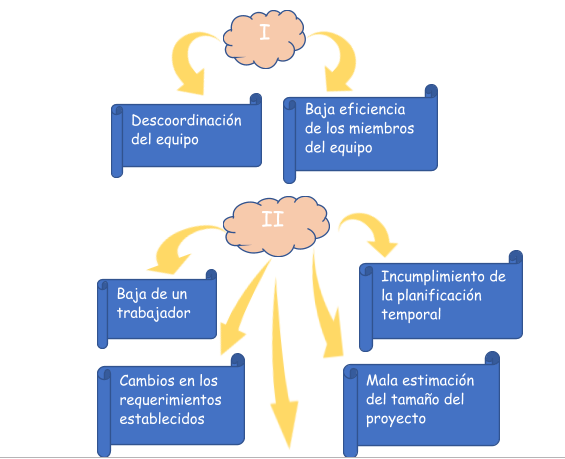
Las correspondientes mediciones se realizarán en **P/h**. Consideramos que hemos trabajado dos horas de media diaria lo cual concuerda con la realidad de forma experimental. A continuación se muestra la **tabla EDT con los esfuerzos.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AE | PLANIFICACIÓN Y GESTIÓN DE RIESGOS | INGENIERÍA | | | CONSTRUCCIÓN Y ADAPTACIÓN | | | EVALUACIÓN DEL PROYECTO | Total |
| Acción |  | Análisis | | Diseño | Codificación | Prueba | Ensamblaje |  |  |
| **VOTACIONES** | Análisis de riesgos  14 p/h | Análisis requisitos del sistema  34 p/h | | Diseño del subsistema Votaciones  16 p/h | Codificación del subsistema Votaciones  32 p/h | Prueba del subsistema Votaciones  22 p/h | Ensamblaje de cada módulo  12 p/H | Evaluar diseño subsistema Votaciones  30 p/h | 128 p/h |
| **GESTIÓN DE USUARIOS** | Diseño del subsistema  Gestión de Usuarios  16 p/h | Codificación del subsistema Gestión de Usuarios  32 p/h | Prueba del subsistema Gestión de Usuarios  22 p/h | Evaluar diseño subsistema Gestión de Usuarios  30 p/H | 128 p/h |
| **ALGORITMOS ALEATORIEDAD** | Diseño del subsistema  Algoritmos de Aleatoriedad  16 p/h | Codificación del subsistema Algoritmos de Aleatoriedad  32 p/h | Prueba del subsistema Algoritmos de Aleatoriedad  22 p/h | Evaluar diseño subsistema Algoritmos de Aleatoriedad  30 p/H | 128 p/h |
| **PROYECTO** | Elaborar Plan de Gestión del Riesgo  14 p/h | Diseño del proyecto  Casos de uso: 63 p/h  Pantallas: 30 p/h  Diagrama de Gantt: 60 p/h  Diagrama de CU: 10 p/h  SRS:16 p/h  PP: 16 p/h | | | Revisiones de Código  1ª Revisión Código: 14 p/h  2ª Revisión Código: 14 p/h  Revisión Definitiva: 12 p/h | | | **Entrega Proyecto**  **(9/05/19)** | 249 |
| Revisión Plan Gestión del Riesgo  2 p/H | Revisión SRS  2 p/H | Revisión PP  2 p/H | | Cierre Código  2 p/h | | | +8  = 257 |
| **Estimación**  **total** | 30 p/h | 36 p/h | 243 | | 240 | | | 90 | 639 |
| **Esfuerzo** | 4,70 | 5,63 | 38 | | 37,59 | | | 14 |  |

# 3. ESTRATEGIA DE GESTIÓN DEL RIESGO

**Autor:** María Juberías

**Supervisa**: Daniel Jiménez



## 3.1 IDENTIFICACIÓN DE RIESGOS DEL PROYECTO

Los **principales riesgos** que pueden afectar al proyecto son:

|  |  |  |
| --- | --- | --- |
| TIPO DE RIESGO | RIESGO | DESCRIPCIÓN |
| NEGOCIO | Falta de financiación | Por diferentes motivos el coste podría ser subestimado, necesitándose así un mayor presupuesto que el previsto. |
| La aplicación no tiene la acogida esperada por el público | El usuario no está conforme con la aplicación, o bien por motivos funcionales o bien por motivos estético |
| PROYECTO | Baja de un trabajador | Si esto llegara a suceder estaríamos obligados a modificar la planificación temporal y la distribución de las tareas de los miembros del equipo |
| Abandono de algún miembro del equipo | El abandono de cualquier integrante provocaría la necesidad de redistribución de las tareas y un cambio en la planificación temporal. |
| Incumplimiento de la planificación temporal | Las estimaciones temporales pueden ser poco precisas, lo que nos puede llevar a necesitar modificar costes. |
| Cambios en los requerimientos establecidos | Conforme se realiza el proyecto pueden surgir la necesidad de nuevas funcionalidades o el cambio parcial o total de las ya existentes, lo cual conlleva una modificación del coste. |
|  | Mala estimación del tamaño del proyecto | Las estimaciones del tamaño pueden ser poco precisas, lo que nos puede llevar a necesitar modificar costes. |
| Descoordinación del equipo | Puede producirse una falta de comprensión o de comunicación entre los miembros del equipo, provocando un ambiente hostil, además de una serie de problemas causados por la falta del intercambio de la información. |
| Baja eficiencia de los miembros del equipo | Algún miembro no participa lo suficiente en el proyecto provocando un retraso y una sobrecarga del resto del equipo |
| TÉCNICOS | Fallos en el acceso a la base de datos | Es necesario una buen funcionamiento del servidor, para que la funcionalidad del programa sea correcta en todo momento |
| Lanzar la aplicación al mercado con un bug | El diseño de la aplicación tiene un defecto importante. |
| Ineficiencia del código | Una vez acabado el código es posible que la aplicación no tenga un funcionamiento más lento del esperado,, siendo necesaria una revisión del código para su optimización. |
| Falta de conocimientos para el desarrollo de la aplicación | Para la implementación de la aplicación se necesitan una serie de tecnologías que no están al alcance de los miembros del equipo por su complejidad |
| LEGALES | Divulgación de datos personales | En esta aplicación los usuarios deben de meter datos, que por la política de privacidad no pueden salir a la luz |

## 3.2 PRIORIZACIÓN DE RIESGOS DEL PROYECTO

A continuación mostramos una clasificación de la prioridad de los riesgos basándonos en el criterio **SQAS-SEI**, en el cual se establecen 5 niveles de prioridad según el tipo de riesgo y el impacto que este pueda llegar a producir en el proyecto. Los riegos serán clasificados según la tabla siguiente:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FRECUENCIA | FRECUENTE | PROBABLE | OCASIONAL | REMOTO | IMPROBABLE |
| SEVERIDAD |
| CATASTRÓFICO | IN | IN | IN | H | M |
| CRÍTICO | IN | IN | H | M | L |
| SERIO | H | H | M | L | T |
| MENOR | M | M | L | T | T |
| INSIGNIFICANTE | M | L | T | T | T |
| LEYENDA | IN. Intolerante | H.Alto | M.Medio | L. Bajo | T. Tolerante |

|  |  |  |  |
| --- | --- | --- | --- |
| RIESGO | FRECUENCIA | IMPACTO | PRIORIDAD |
| Falta de financiación | REMOTO | CRÍTICO | MEDIA |
| La aplicación no tiene la acogida esperada por el público | PROBABLE | MENOR | MEDIA |
| Baja de un trabajador | OCASIONAL | CRÍTICO | ALTA |
| Abandono de algún miembro del equipo | IMPROBABLE | CRÍTICO | BAJA |
| Incumplimiento de la planificación temporal | PROBABLE | SERIO | ALTA |
| Cambios en los requerimientos establecidos | OCASIONAL | CRÍTICO | ALTO |
| Mala estimación del tamaño del proyecto | PROBABLE | SERIO | ALTA |
| Descoordinación del equipo | OCASIONAL | CATASTRÓFICO | INTOLERABLE |
| Baja eficiencia de los miembros del equipo | PROBABLE | CRÍTICO | INTOLERABLE |
| Fallos en el acceso a la base de datos | REMOTO | CRÍTICO | MEDIO |
| Lanzar la aplicación al mercado con un bug | REMOTO | CRÍTICO | MEDIO |
| Ineficiencia del código | OCASIONAL | SERIO | MEDIO |
| Falta de conocimientos para el desarrollo de la aplicación | FRECUENTE | SERIO | ALTO |
| Divulgación de datos personales | IMPROBABLE | CATASTRÓFICO | MEDIA |

Por ello los niveles de prioridad en orden son los siguientes:

|  |  |
| --- | --- |
| NIVEL DE PRIORIDAD 1: INTOLERABLE | [Descoordinación del equipo](#Descoordinacion) |
| Baja eficiencia de los miembros del equipo |

|  |  |
| --- | --- |
| NIVEL DE PRIORIDAD 2: ALTO | [Baja de un trabajador](#BajaTrabajador) |
| Incumplimiento de la planificación temporal |
| Cambios en los requerimientos establecidos |
| Mala estimación del tamaño del proyecto |
| Falta de conocimientos para el desarrollo de la aplicación |

|  |  |
| --- | --- |
| NIVEL DE PRIORIDAD 3: MEDIO | [Falta de financiación](#Financiacion) |
| La aplicación no tiene la acogida esperada por el público |
| Fallos en el acceso a la base de datos |
| Lanzar la aplicación al mercado con un bug |
| Ineficiencia del código |
| Divulgación de datos personales |

|  |  |
| --- | --- |
| NIVEL DE PRIORIDAD 4: BAJO | [Abandono de algún miembro del equipo](#AbandonoMiembro) |

## 3.3 PLAN DE GESTIÓN DE RIESGO

Presentamos a continuación una serie de tablas de gestión de los riesgos establecidos con previamente, indicando las acciones necesarias para prevenirlos, supervisarlos y en caso de que se den un **plan de contingencia** para poder solucionar o mitigar el riesgo.

|  |  |
| --- | --- |
| **NIVEL DE PRIORIDAD 1**: | |
| RIESGO: | Descoordinación del equipo |
| PREVENCIÓN | Las tareas se distribuirán de forma equitativa, dejando a los miembros una elección prioritaria basada en gustos o en presentaciones realizadas en clase con prioridad. |
| SUPERVISIÓN | Todas las partes del equipo estarán comunicadas y el intercambio de información será constante. |
| PLAN DE CONTINGENCIA | Reuniones del equipo e intercambio de miembros en aquellas tareas que sean necesarias o que no hayan resultado exitosas. |
| RIESGO: | Baja eficiencia de los miembros del equipo |
| PREVENCIÓN | Cada miembro podrá elegir que parte realizar según sus capacidades y dotes para que durante su realización esté motivado. |
| SUPERVISIÓN | El líder estará atento de que todos los componentes estén dando su mejor versión. |
| PLAN DE CONTINGENCIA | Conversar con aquellas personas involucradas en la parte problemática intentando animarlas y llegar a una solución que mejore su estado. |

|  |  |
| --- | --- |
| **NIVEL DE PRIORIDAD 2:** | |
| RIESGO: | Baja de un trabajador |
| PREVENCIÓN | No se puede prever cuando un trabajador enfermará |
| SUPERVISIÓN | Todos los componentes del equipo deben supervisar a sus compañeros para que se tenga constancia del trabajo que ha sido realizado en cada periodo de tiempo dado. |
| PLAN DE CONTINGENCIA | Reorganizar la planificación temporal y redistribuir al resto del equipo las tareas a realizar por la persona de baja. |
| RIESGO: | Incumplimiento de la planificación temporal |
| PREVENCIÓN | La estimación temporal debe ser lo más realista posible. |
| SUPERVISIÓN | Se debe llevar constancia del cumplimiento de los periodos temporales. |
| PLAN DE CONTINGENCIA | Reajustar los periodos necesarios, e intentar ceñirse más a la nueva planificación. |
| RIESGO: | Cambios en los requerimientos establecidos |
| PREVENCIÓN | Todos los miembros del equipo  deberán revisar y comprobar los requisitos establecidos. |
| SUPERVISIÓN | Comunicación constante entre los miembros del equipo por si surge una nueva necesidad en el proyecto. |
| PLAN DE CONTINGENCIA | Rehacer especificación de requisitos y todas las partes del proyecto que requieran cambios. |
| RIESGO: | Mala estimación del tamaño del proyecto |
| PREVENCIÓN | La estimación del tamaño debe ser lo más realista posible. |
| SUPERVISIÓN | Controlar que se van cumpliendo los plazos y las estimaciones realizadas. |
| PLAN DE CONTINGENCIA | Reajustar aquello que sea necesario e intentar ceñirse más a la nueva planificación. |
| RIESGO: | Falta de conocimientos para el desarrollo de la aplicación |
| PREVENCIÓN | Evitar implementaciones al proyecto que se salgan de nuestro conocimiento, cambiándolas por alternativas más simples o directamente eliminándolas si no son estrictamente necesarias. |
| SUPERVISIÓN | Se harán reuniones periódicas para revisar futuras implementaciones y comprobar si algún miembro es capaz de realizarla. |
| PLAN DE CONTINGENCIA | Documentarnos sobre las herramientas necesarias para realizar dicha tarea y aprender a utilizarlas. |

|  |  |
| --- | --- |
| **NIVEL DE PRIORIDAD 3:** | |
| RIESGO: | Falta de financiación |
| PREVENCIÓN | Prever de forma correcta los costes e intentar conseguir la financiación necesaria. |
| SUPERVISIÓN | Controlar que los gastos van ocurriendo de la misma forma en la que han sido previstos. |
| PLAN DE CONTINGENCIA | Pedir un préstamo a proveedores o a alguna entidad bancaria |
| RIESGO: | La aplicación no tiene la acogida esperada por el público |
| PREVENCIÓN | Hacer un estudio previo del público al que va dirigida principalmente la aplicación, diseñando la aplicación de forma atractiva. |
| SUPERVISIÓN | Controlar que el diseño sea atractivo. |
| PLAN DE CONTINGENCIA | Intentar mejorar aquellos aspectos que no sean lo suficiente atractivos, proponiendo cambios originales y llamativos. |
| RIESGO: | Fallos en el acceso a la base de datos |
| PREVENCIÓN | Tener una copia de seguridad. |
| SUPERVISIÓN | Comprobar que la copia se realiza de forma adecuada por el sistema, así como el óptimo funcionamiento del servidor.. |
| PLAN DE CONTINGENCIA | Utilizar otro servidor. |
| RIESGO: | Lanzar la aplicación al mercado con un bug |
| PREVENCIÓN | Elaborar un plan de proyecto detallado, realizando gran cantidad de pruebas durante la implementación de esta. |
| SUPERVISIÓN | Revisar la aplicación intentando encontrar la mayor cantidad de errores posibles antes de su lanzamiento en el mercado. |
| PLAN DE CONTINGENCIA | Una vez encontrado un error, realizar una nueva versión que lo subsane. |
| RIESGO: | Ineficiencia del código |
| PREVENCIÓN | Planificar la arquitectura del código para buscar la alternativa más eficiente. |
| SUPERVISIÓN | Analizar el tiempo y memoria utilizados mediante pruebas cada vez que se termine un subsistema funcional del proyecto. |
| PLAN DE CONTINGENCIA | Modificar las partes del código que provocan una lentitud no deseada. |
| RIESGO: | Divulgación de datos personales |
| PREVENCIÓN | Tener en cuenta la vigente legislación a la hora de la implementación del proyecto, especialmente respetando la protección de datos. |
| SUPERVISIÓN | Almacenar con un cifrado seguro los datos. |
| PLAN DE CONTINGENCIA | Cambiar las partes del proyecto que incumplan la ley vigente y mejorar las medidas de seguridad. |

|  |  |
| --- | --- |
| **NIVEL DE PRIORIDAD 4:** | |
| RIESGO: | Abandono de algún miembro del equipo |
| PREVENCIÓN | Exigir un compromiso a los miembros del equipo y exigir que abandone únicamente por problemas de fuerza mayor. |
| SUPERVISIÓN | Mantener a todos los miembros del equipo informados en todo momento sobre aquellas tareas que se van realizando. |
| PLAN DE CONTINGENCIA | Realizar una nueva planificación temporal y volver a distribuir las tareas entre los miembros del equipo. |

## 3.4 PLANIFICACIÓN TEMPORAL DEL CONTROL DE RIESGO

Los **riesgos** serán abordados por los miembros del equipo cada vez que surjan. Se organizarán **reuniones periódicas** en las que se pondrán en común los problemas que hayan surgido y las posibles soluciones a ellos, con el objetivo de obtener una solución de forma rápida y eficiente que subsane el problema. Dichas reuniones se realizarán al principio de cada semana y tendrán una duración de 15-20 min.

**Resumen**

En esta parte del documento han sido descritos los posibles riesgos de **DeciDT**, realizando un análisis de frecuencia e impacto y siendo clasificados según prioridad. Cada uno de los riesgos contiene una serie de pasos a seguir para evitarlos, y en el caso de que se lleguen a dar cuentan con un plan de contingencia para eliminar el riesgo, o por lo menos mitigarlo, con el objetivo de conseguir superar las posibles dificultades que vayan surgiendo y de sacar adelante el proyecto de manera exitosa.

# 4. PLANIFICACIÓN TEMPORAL

**Autor**: Alejandro Ramírez

**Supervisa**: Laura Marco

## 4.1 ESTRUCTURA DE DESCOMPOSICIÓN DEL TRABAJO

Una de las normas implícitas del desarrollo de nuestro proyecto es la **justa repartición del trabajo**, tanto en esfuerzo como en tiempo. La planificación temporal que hemos elaborado pretende acercarse lo máximo posible a ese ideal de justicia, partiendo de la base de que siempre habrá miembros del equipo que trabajarán más.

A la hora de descomponer el trabajo, primero realizamos una **partición modular**, basada en dividir el proyecto en módulos. Una vez que estos están establecidos, experimentan una **partición vertical**. En esta descomposición, cada módulo se trocea en sus diversas funciones.

A continuación mostramos la estructura de la descomposición del trabajo en dos tablas, pues en la segunda se especifica la descomposición del módulo Proyecto en dos submódulos: SRS y PP.

|  |  |
| --- | --- |
| *MÓDULOS* | *ACTIVIDADES* |
| VOTACIONES | * Diseño de votaciones * Evaluación diseño de votaciones * Codificación de votaciones * Revisión código de votaciones * Prueba código de votaciones |
| GESTIÓN DE USUARIOS | * Diseño de gestión de usuarios * Evaluación diseño de gestión de usuarios * Codificación de gestión de usuarios * Revisión código de gestión de usuarios * Prueba código de gestión de usuarios |
| ALGORITMOS ALEATORIOS | * Diseño de algoritmos aleatorios * Evaluación diseño de algoritmos aleatorios * Codificación de algoritmos aleatorios * Revisión código de algoritmos aleatorios * Prueba código de algoritmos aleatorios |
| PROYECTO | * Análisis del proyecto * Diseño del proyecto * Diseño SRS * Diseño PP * Cierre código * Entrega proyecto |

|  |  |
| --- | --- |
| MÓDULO PROYECTO | |
| *SUBMÓDULOS* | *ACTIVIDADES* |
| SRS | * Redacción de la introducción * Descripción general proyecto * Identificación y descripción casos de uso * Identificación y descripción actores * Diseño y descripción de pantallas * Requerimientos técnicos * Requerimientos funcionales * Revisión del documento |
| PP | * Presentación del proyecto * Estimaciones y documentación previa * Desarrollo plan de gestión del riesgo * Planificación temporal * Recursos del proyecto |

A continuación mostramos la tabla EDT:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AE | PLANIFICACIÓN Y GESTIÓN DE RIESGOS | INGENIERÍA | | | CONSTRUCCIÓN Y ADAPTACIÓN | | | EVALUACIÓN DEL PROYECTO |
| Acción |  | Análisis | | Diseño | Codificación | Prueba | Ensamblaje |  |
| **VOTACIONES** | Análisis de riesgos  (27/11/18--4/12/18) | Análisis requisitos del sistema  (12/11/18-16/11/18)  (19/11/18-30/11/18) | | Diseño del subsistema Votaciones  (23/01/19—1/02/19) | Codificación del subsistema Votaciones  (17/02/19—3/04/19) | Prueba del subsistema Votaciones  (12/04/19—16/04/19)  (25/04/19—30/04/19) | Ensamblaje de cada módulo  (7/05/19—12/05/19) | Evaluar diseño subsistema Votaciones  (2/02/19—16/02/19) |
| **GESTIÓN DE USUARIOS** | Diseño del subsistema  Gestión de Usuarios  (23/01/19—1/02/19) | Codificación del subsistema Gestión de Usuarios  (17/02/19—3/04/19) | Prueba del subsistema Gestión de Usuarios  (12/04/19—16/04/19)  (25/04/19—30/04/19) | Evaluar diseño subsistema Gestión de Usuarios  (2/02/19—16/02/19) |
| **ALGORITMOS ALEATORIEDAD** | Diseño del subsistema  Algoritmos de Aleatoriedad  (23/01/19—1/02/19) | Codificación del subsistema Algoritmos de Aleatoriedad  (17/02/19—3/04/19) | Prueba del subsistema Algoritmos de Aleatoriedad  (12/04/19—16/04/19)  (25/04/19—30/04/19) | Evaluar diseño subsistema Algoritmos de Aleatoriedad  (2/02/19—16/02/19) |
| **PROYECTO** | Elaborar Plan de Gestión del Riesgo  (13/12/18—19/12/18) | Diseño del proyecto  (12/11/18—23/12/18)  Casos de uso: (13/11/18—4/12/18)  Pantallas: (21/11/18—4/12/18)  Diagrama de Gantt: (23/11/18—23/12/18)  Diagrama de CU: (13/11/18—17/11/18)  SRS: (13/12/18—21/12/18)  PP: (13/12/18—21/12/18) | | | Revisiones de Código  (4/04/19—6/05/19)  1ª Revisión Código: (4/04/19—11/04/19)  2ª Revisión Código: (17/04/19—24/04/19)  Revisión Definitiva: (1/05/19—6/05/19) | | | Entrega Proyecto  (9/05/19) |
| Revisión Plan Gestión del Riesgo  (20/12/18—21/12/18) | Revisión SRS  (5/12/18—8/12/18)  (20/12/18—21/12/18) | Revisión PP  (5/12/18—8/12/18)  (20/12/18—21/12/18) | | Cierre Código  (7/05/19) | | |

## 4.2 DIAGRAMA DE GANTT

Adjuntamos el diagrama de Gantt como archivo *.gan:* [*Diagrama de Gantt*](file:///C:\Users\laura\PROYECTO%20INICIO\Anexos%20PT\Planificacion%20Temporal.gan)

Para facilitar su comprensión, también adjuntamos la red de tareas como imagen .*jpeg,* elaborada con la técnica PERT[: *Imagen PERT*](file:///C:\Users\laura\PROYECTO%20INICIO\Anexos%20PT\DIAGRAMA%20PERT.jpg)

Además, ambos documentos se pueden encontrar en los anexos, junto con un pdf que ofrece más detalles de la planificación temporal.

## 4.3 GESTIÓN DE RECURSOS

|  |  |  |
| --- | --- | --- |
|  | DESARROLLO | CLIENTE |
| HARDWARE | * Ordenadores de los laboratorios * Ordenadores personales | * Teléfonos móviles |
| SOFTWARE | * Modelio, Gantt Project * Gimp, Proto.io * Microsoft Word, LATEX * Slack, Google Drive * Eclipse, Android Studio | * Android KitKat 4.4 o posterior |

# 5. RECURSOS DEL PROYECTO

**Autor:** Mario Jiménez

**Supervisa**: Miguel Macicior

## 5.1 PERSONAL

        El equipo de DeciDT está compuesto por 7 personas. Dentro de estas 7 personas hay una subdivisión por parejas en la que uno realiza una tarea y el otro la revisa; excepto en la tarea de casos de uso, que por acuerdo se realizó en un grupo de tres. En concreto, en este primer cuatrimestre hemos llevado a cabo **tareas de documentación** y diseño de la interfaz y de los sistemas, entre otras. No obstante, en el segundo cuatrimestre se prevé que habrás subdivisiones en grupos más extensos. Por otro lado, dado que nuestra **comunicación era horizontal**, realmente no ha habido un primus inter pares sino que todos teníamos el mismo voto; no obstante, destacamos la función de María como encargada de participar en reuniones con Borja.

## 5.2 HARDWARE Y SOFTWARE

### **HARDWARE**

**-Hardware de desarrollo**: el desarrollo se llevará a cabo en los **ordenadores personales** de los integrantes del equipo.

-**Hardware de administración**: se necesita un **servidor externo** para poder gestionar de forma centralizada todos los datos relacionados con las cuentas de los usuarios, preguntas disponibles, encuestas, etc.

-**Hardware del cliente**: simplemente el cliente necesita un **Smartphone Android** para instalar y ejecutar la aplicación. (No se descarta que en otro proyecto paralelo se compatibilice para dispositivos Apple).

### **SOFTWARE**

Software de desarrollo:

**Android Studio**: IDE oficial para la plataforma Android.

**MySQL**: Herramienta para la gestión de bases de datos.

**GanttProject**: Herramienta para realizar diagramas de planificación temporal.

**Trello**: para monitorizar la lista de tareas pendientes.

**Modelio**: permite la creación de diagramas de casos de uso, de actividad, entre otros.

**Microsoft Word**: editor de texto para documentación. (Conversor a pdf también).

**GitHub**: para gestionar la creación de código simultáneo.

**Eclipse**: Entorno de desarrollo para trabajar en Java.

**Proto.io**: Página web para la creación de prototipos de interfaces.

**Discord**: utilizado para la comunicación entre los componentes del grupo.

**Google Drive**: herramienta de almacenamiento de documentos online que nos permite interactuar en ellos de manera simultánea.

**Software del cliente**: no se demanda software especializado, simplemente un dispositivo Android y Google Play Store para instalar dicha app.

# 6. Plan de GCS / Mecanismos de Gestión y Control

**Autor**: Eduardo López

**Supervisa**: Alejandro Ramírez

## 6.1 Garantía de calidad y control

Para que el proyecto DeciDT sea un éxito,  es muy importante tomar en cuenta la **visión crítica de los usuarios** de la aplicación, por lo que hemos decidido controlar la calidad del programa a través de un sistema de retroalimentación (o **feedback**), para así tratar de cumplir con la mayor cantidad posible de exigencias de los usuarios.

Cuando se realicen cambios al proyecto, se deberá **generar un IEP**. Esto es un documento donde se recoge el estado del proyecto tras la realización del cambio. El fin de este es que el equipo esté informado de qué sucede.

Cada integrante del equipo es responsable tanto de realizar un número de actividades relacionadas con el desarrollo del programa, y también de revisar que estén bien hechas otras que no ha realizado él mismo. Esto genera un mayor **involucramiento del equipo** con todas las actividades relacionadas con la app y por lo tanto, conseguimos un producto de mayor calidad.

### **Conceptos Clave**

* **ECS**: elementos de configuración del software.
* **PGC**: Plan de Gestión de Configuración
* **GCS**: Gestión de configuración del software
* **CCC**: Comité de control de configuración
* **FPC**: formulario de petición de cambio
* **IEP**: Informes del estado del proyecto
* **Líneas Base**: Especificación o producto que se ha revisado formalmente y sobre el que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos de control de cambio.

## 6.2 Gestión de la configuración

La organización de la GCS está estructurada a partir de un **comité de control de cambios**. Esta mesa estará formada por todos los integrantes del proyecto (7 miembros), de entre los cuales se repartirán diferentes labores que son indispensables para el control de la configuración. Estas son:

|  |  |
| --- | --- |
| **Labores** | **Integrantes** |
| Arquitectos de la Aplicación | Laura Marco  Mario Jiménez |
| Arquitectos técnicos | Eduardo López  Daniel Jiménez |
| Directores Económicos | María Juberías  Alejandro Ramírez |
| Gestor de Versiones | Miguel Macicior |

La **comunicación** entre el equipo es indispensable para asegurarnos de que todos estén cumpliendo con su labor. Por eso cada integrante de las distintas labores del comité se encargará de remitir sus resultados. Usamos WhatsApp como plataforma principal de comunicación para mantener al comité informado de las diferentes situaciones que ocurran dentro del desarrollo del programa. A continuación se da una breve descripción de las labores del comité de control de cambios

.

|  |  |
| --- | --- |
| **Labores** | **Descripción** |
| Arquitectos de la Aplicación | Se encargan de identificar Las ECS de la app a nivel general. Desarrollan especificaciones funcionales a nivel general |
| Arquitectos técnicos | Se encargan de identificar los ECS de la app a nivel de programación. |
| Directores Económicos | Se encargan de analizar los cambios planteados en las reuniones del comité y reportar su posible coste de implementación. |
| Gestor de Versiones | Se encargará de crear una carpeta compartida con todos los documentos del proyecto y todas las versiones clasificadas ordenadamente en carpetas. |

Las reuniones del comité serán responsivas a las dificultades que se puedan encontrar en el desarrollo o implementación de la app, por lo que no podremos contar con reuniones periódicas.

Cualquier miembro del equipo de DeciDT tiene la posibilidad de **solicitar un cambio** tanto a los documentos del proyecto, como al código de la app. Para hacer esto tendrá que rellenar un formulario de petición de cambio, que tendrá el siguiente formato:

|  |
| --- |
| Formulario De Petición de Cambio:   * Nombre del solicitante:............................................................................................. * Nivel de Urgencia:.................................................................................................... * Descripción del Cambio:..........................................................................................   ……………………………………………………………………………………………..  ……………………………………………………………………………………………..  ……………………………………………………………………………………………..  …………………………………………………………………………………………….. |

Después de haber entregado el **FPC**, el comité se reunirá para hacer una votación para decidir si se realizará el cambio. Esta votación dará alguno de los siguientes resultados posibles:

* **Se realiza el cambio**. Para que el cambio se realice tendrá que contar con una mayoría de al menos 5 integrantes.
* **Se rechaza el cambio**. Si la votación da como resultado que solo tres o menos integrantes del equipo votan a favor del cambio, el cambio será rechazado.
* **Se discute y se hace una segunda votación**. Si son 4 los que votan a favor del cambio, se abrirá una ventana para discutir y después se hará una segunda votación. En esta segunda votación se necesitarán 4 votos a favor y no 5 para realizar el cambio.

## 6.3 Control de Versiones

Se planea usar **Gitlab** para gestionar el control de versiones. También se utilizará un software de almacenamiento en la nube como **Google Drive**, así como algunos softwares de comunicación como el correo electrónico y WhatsApp. De esta forma, se creará una carpeta compartida con todos los documentos del proyecto clasificados ordenadamente en carpetas.

Para realizar cambios en los documentos o en el código de la app, se aplicará el protocolo descrito anteriormente. Una vez aprobados los cambios, El **gestor de versiones** modificará los ficheros afectados en la carpeta compartida de Google Drive, manteniendo las versiones anteriores, y registrando el cambio en el apartado correspondiente dentro del fichero. Finalmente, el responsable de implementar el cambio deberá registrar el informe en una carpeta específica de Informes de Cambios Implementados.

## 6.4 Líneas Base

**Línea base 0: Objetivo del proyecto**

DeciDT es una aplicación que pretende ayudar a los usuarios a realizar decisiones, tanto simples como complejas. Las decisiones “simples” se resolverán a través de diferentes juegos de azar, pero las complejas involucrarán a otros usuarios, donde se podrá pedir consejos, abrir votaciones o encuestas, entre otras opciones.  Pretende para ello crear una comunidad en la que los usuarios puedan contar con ayuda de otros para guiarlos al enfrentarse con una decisión difícil.

**Línea base 1: Forma del proyecto y dispositivos**

DeciDT está planteada inicialmente como una **app móvil**, por lo que se implementará a los principales **sistemas operativos móviles** (Android y iOS). Podrá descargarse a través de Google Play Store y la Apple App Store.

**Línea base 2: Movilidad y Accesibilidad**

Para utilizar los servicios de toma de decisiones complejas de DeciDT,  se necesitará uso de **datos móviles**, pero para los juegos de azar, no harán falta.

**Línea base 3: Acceso a la app**

Para usar el servicio de realizar decisiones complejas, el usuario tendrá que **crear una cuenta en DeciDT**, pero los usuarios podrán usar los juegos de azar y participar en decisiones complejas a las que sean invitados sin necesidad de registrarse.

**Línea base 4: Relación entre app y usuarios**

DeciDT se encargará de crear una **base de datos con las decisiones complejas,** así usuarios con alguna nueva podrán buscar problemas similares a los que tienen  dentro de las decisiones resueltas.

El usuario le dará un feedback a la aplicación cuando se cierre alguna decisión compleja ya resuelta.

**Línea base 5: Programación de la aplicación**

DeciDT estará programado en **Java**. Se tiene pensado utilizar eclipse para la programación.

El servidor de DeciDT estará programado en **JavaScript Node**.

**Línea base 6: Protección de Datos**

DeciDT se compromete a mantener la privacidad de los datos de los usuarios recabados por la aplicación, así como a no comercializarlos a terceros.

Los integrantes de  DeciDT se comprometen a no utilizar los datos de los usuarios de la aplicación para fines ajenos al desarrollo y mejora de DeciDT.

# 7. GESTIÓN DE EQUIPO

**Autor:** Daniel Jiménez

**Supervisa**: Eduardo López

## 7.1. Estructura del equipo

Estudiemos en primer lugar los factores de **Mantei** para determinar la estructura del equipo:

|  |  |
| --- | --- |
| **DIFICULTAD DEL PROBLEMA** | La dificultad del proyecto es media, ya que las funciones a realizar están claras y son de una dificultad media. |
| **TAMAÑO DEL PROBLEMA** | El tamaño es medio, puesto que el proyecto no cuenta con un número de funciones relativamente alto. |
| **DURACIÓN DEL EQUIPO** | Salvo abandonos o bajas imprevistas, el equipo será el mismo desde el inicio hasta la finalización del proyecto. |
| **MODULARIDAD** | La modularidad del programa es media-baja. Uno de los subsistemas, el de algoritmos de aleatoriedad, es completamente independiente al resto, sin embargo, gestión de usuarios y votaciones están relacionadas. |
| **CALIDAD Y FIABILIDAD DEL SISTEMA A CONSTRUIR** | El proyecto ha de ser funcional desde su finalización y cumplir los requisitos de calidad establecidos con una fiabilidad alta. El número de defectos debe ser mínimo, por ello deben encontrarse la mayor cantidad de errores posibles antes de la entrega. |
| **FECHA DE ENTREGA** | Contamos para finalizar el proyecto hasta junio de 2019. Esta fecha es estricta. |
| **COMUNICACIÓN** | Debido a la modularidad, esta deberá ser media-alta, pues aunque el subsistema independiente, el de algoritmos de aleatoriedad, no requiere comunicación, entre los subgrupos encargados de los otros dos subsistemas necesitan una alta comunicación. |

Teniendo en cuenta todos estos factores, y en especial por la **necesidad de una comunicación y fiabilidad altas**, nos hemos decantado por un **modelo descentralizado democrático** con una ligera modificación, la cual consiste en formar varios subgrupos que se encargarán cada uno de un subsistema concreto.

## 7.2 Informes de gestión

Procederemos a detallar las funciones a realizar en cada subsistema:

|  |  |
| --- | --- |
| **SUBSISTEMA DE GESTIÓN DE USUARIOS** | Engloba las funciones relativas a los usuarios.   * Añadir usuario: Si un usuario lo desea, debe poder registrar una nueva cuenta si esta no se encuentra ya almacenada en la base de datos. * Log in/out: Los usuarios tienen que poder iniciar y cerrar sesión en la aplicación de móviles. * Almacenar datos: Si los datos de un usuario se han visto modificados, esta nueva información debe guardarse en la base de datos. * Agregar amigos: Las cuentas de los usuarios contarán con una lista de amigos en las que podrán añadir a los usuarios que deseen. * Ver datos guardados: Los usuarios han de tener la opción de ver los datos de su cuenta (lista de amigos, puntos ganados por participar en votaciones, votaciones propias, comentarios realizados, etc). * Mostrar notificaciones: La aplicación mostrará a los usuarios novedades interesantes como puede ser una nueva votación privada en curso. |
| **SUBSISTEMA DE VOTACIONES** | Engloba todas las funcionalidades que conciernen a las votaciones:   * Añadir votación: A los usuarios se les ha de ofrecer la opción de crear una nueva votación que se guardará en la base de datos. * Añadir voto y comentario: Los usuarios deben poder participar en una votación en curso dejando su voto y comentario, que tendrán que ser almacenados en la base de datos. * Registrar feedback: Los usuarios han de tener la opción de dar un voto positivo o negativo a comentarios de votaciones en curso. * Ver votaciones en curso: Los usuarios, si lo requieren, tienen que poder ver las votaciones públicas abiertas para participar en ellas. |
| **SUBSISTEMA DE ALGORITMOS DE ALEATORIEDAD** | Engloba las funciones que involucren aleatoriedad:   * Tomar decisión aleatoria: Los usuarios deben poder de introducir un determinado número de opciones y que el sistema les escoja y muestre una. * Organizar torneo : Los usuarios han de tener la opción de introducir los datos de un torneo: equipos y fases, y el sistema debe realizarles y mostrarles una organización aleatoria. |

Por último, dejamos detallada la estructura de los subgrupos encargados del desarrollo de cada subsistema:

|  |  |  |
| --- | --- | --- |
|  | AUTORES | REVISIÓN |
| SUBSISTEMA DE GESTIÓN DE USUARIOS | **Alejandro Ramírez**  Miguel Macicior | Laura Marco  Mario Jiménez |
| SUBSISTEMA DE VOTACIONES | **Daniel Jiménez**  Mario Jiménez  Laura Marco | Eduardo López  Alejandro Ramírez  María Juberías |
| SUBSISTEMA DE ALGORITMOS DE ALEATORIEDAD | **María Juberías**  Eduardo López | Daniel Jiménez  Miguel Macicior |

**Abc**: Supervisores

En la tabla se ha destacado el nombre de un miembro de cada subgrupo cuya ocupación será la de supervisor. Estos, además de programar como el resto de su subgrupo, se encargarán de la integración de su subsistema con el resto, así como de llevar a cabo rápidas comprobaciones del código en busca de posibles errores, procurando así que, una vez llegada la revisión final, los cambios necesarios sean mínimos.

Daniel Jiménez

Mario Jiménez

Laura Marco

Votaciones

Gestión de usuarios

Alejandro Ramírez

Miguel Macicior

Algoritmos de Aleatoriedad

María Juberías

Eduardo López

***DeciDT***

Supervisores

# 8. ANEXOS Y REFERENCIAS

## **ANEXO**

En los anexos se encuentran la red de tareas en formato jpeg y el proyecto de Gantt Project para la planificación temporal.

## **REFERENCIAS**

Como punto de partida, hemos tomado los documentos de la entrega Enjoy Mad y Complutensers, ofrecidos en el campus virtual.

Hemos usado estas páginas como fuente de información:

[1] <https://www.elperiodico.com/es/extra/20180220/veintidos-millones-espanoles-apps-6635109>

[2] El Proceso Unificado de Desarrollo de Software ([ISBN 84-7829-036-2](https://es.wikipedia.org/wiki/Especial:FuentesDeLibros/8478290362)) y fue publicado en 1999 por [Ivar Jacobson](https://es.wikipedia.org/wiki/Ivar_Jacobson), [Grady Booch](https://es.wikipedia.org/wiki/Grady_Booch) y [James Rumbaugh](https://es.wikipedia.org/wiki/James_Rumbaugh)