**3.2 Suposiciones, dependencias y restricciones**

Que las herramientas proporcionadas por los profesores funcionan, como así hemos comprobado ejecutando el código del Buscaminas el cuál tiene una funcionalidad mínima que iremos ampliando, o el Microsoft Office que nos ayudará a realizar algunos diagramas y organizar tareas entre otras cosas.

También que el ordenador que se nos proporciona para la presentación funcione correctamente, así como los ordenadores de clase para poder trabajar desde ellos en las horas de clase que se nos dejan para realizar la práctica.

Además de los ordenadores del aula también hemos necesitado acceder a myApps para utilizar algunas herramientas que se nos proporcionan en este entorno como puede ser Microsoft Project.

También que estas herramientas son las adecuadas, como los estándares a seguir para realizar este proyecto.

En cuanto a los acontecimientos externos de los cuáles el proyecto depende podríamos considerar:

-La estabilidad de GitHub, WhatsApp, Skype y demás herramientas que hemos utilizado para comunicarnos y que su funcionamiento no está en nuestra mano. Ya que la comunicación fluida entre los miembros del grupo es primordial para la realización de este proyecto

-Ya que es un proyecto académico, que las herramientas sean gratuitas y no necesitemos presupuesto para conseguirlas

-Que el personal esté en condiciones para realizar su parte

Restricciones bajo las cuales el proyecto va a ser guiado:

-No menos de 30 páginas de extensión

-El grupo debe estar formado por 3-5 personas

-Fecha límite para entregar el proyecto es el 25 de abril a las 23:55h

-Que el diagrama de Gant tenga entre 15 y 20 actividades

-Que el diagrama de Pert tenga un mínimo de 20 actividades

-Relacionado con el código que se nos pide mejorar podemos considerar las siguientes restricciones:

• Añadir una opción para poder reiniciar el juego en cualquier momento y sin tener que reiniciar el programa.

• Mostrar en todo momento la cantidad de minas que quedan por descubrir, así como el tiempo que llevamos jugando en la partida actual.

• Crear diferentes niveles de dificultad:

- Principiante (tablero 10x10 y 10 minas)

-Intermedio (tablero 16x16 y 40 minas)

- Experto (tablero 32x16 y 99 minas)

-Personalizado (permite definir el tamaño del tablero y la cantidad de minas)

• Guardar los 10 mejores tiempos en cada dificultad (sin contar Personalizado) • En el mensaje al ganar, mostrar junto con el tiempo un campo para añadir un nombre de jugador y dar la opción de guardar el tiempo o no.

• Permitir guardar la partida actual (estado del tablero y tiempo) en un fichero, así como poder recuperar una partida guardada a partir de un fichero.

• Crear un menú que nos permita acceder a las diferentes opciones del juego.

**3.3 Gestión de riesgos**

Podemos identificar varios riesgos:

-Al modificar código, como cada vez que programamos, corremos el riesgo de perder funcionalidad.

-Que al subir cada uno nuestra parte a GitHub haya conflictos.

-Que al ir avanzando en el proyecto nos demos cuenta de nuevas actividades y haya que introducir en los diferentes diagramas con los cambios que ello conlleva.

-Al ser un proyecto académico no corremos riegos con el dinero, pero si con el tiempo empleado en cada actividad por eso hay que planificarlo bien.

-Que en la planificación y distribución de tareas hayamos dedicado poco tiempo a una actividad determinada y no nos de tiempo a realizarla y haya que dedicarle más tiempo. Al revés también, si dedicamos demasiado tiempo a una actividad que no necesita tanto tiempo, corremos el riesgo de no realizar actividades posteriores por falta de tiempo.

Para intentar evitar estos riesgos hay que tener una planificación adecuada y que la comunicación del grupo sea fluida para atajar los problemas a tiempo. Así como el innegociable esfuerzo de cada uno de los miembros para entregar todo a tiempo según lo hemos planificado.

A la hora de distribuir el trabajo es importante que nos ayudemos unos a otros en el caso de que uno acabe su tarea, ya que así avanzaremos más rápidamente y no corremos el riesgo de ir justos con los plazos de entrega.

**3.4 Mecanismos de supervisión y control**

-Para ir controlando el desarrollo del proyecto y sus distintas actividades y paquetes de trabajo nos hemos basado en GitHub y Whatsapp. Mediante estas dos plataformas hemos encontrado un gran mecanismo de control, ya que a GitHub cada miembro subía su parte y por Whatsapp lo que hacíamos era avisar de que habíamos actualizado nuestra parte. Así podíamos enterarnos de cambios lo más rápido posible y ver cómo iba avanzando el proyecto.

-Las herramientas principales que hemos utilizado han sido Word y Microsoft Project. Con el primero creamos los informes y los documentos finales y con el Microsoft Project generamos documentos con la gestión de tareas y diagramas. Cada uno lo genera en su propio formato que nosotros sabemos gestionar sin problemas.

-Una vez hemos terminado la parte de código, el encargado de esta parte mediante la herramienta Eclipse, que es un entorno de desarrollo para programar en diferentes lenguajes, genera el ejecutable entregable cuya extensión es .jar.

**3.5 Plan de personal**

Para llevar a cabo este proyecto los profesores de la asignatura nos recomendaron grupos de entre 3 y 5 personas, y nos ayudaron a completar el equipo de trabajo el cual está formado finalmente por cuatro personas

En cuanto al nivel de habilidad, se requiere un mínimo de conocimiento de Java por parte de los cuatro componentes del grupo para entender lo que se pide e interpretar código.

También saber manejar el Microsoft Project para realizar los diagramas y gestionar las tareas.

Así cómo saber cómo funciona GitHub para que cada miembro del grupo suba su trabajo y éste sea visible para el resto del grupo.

Para trabajar con el código proporcionado por los profesores de la asignatura, tras consensuarlo entre los componentes del grupo, hemos decidido utilizar Eclipse que tiene una interfaz bastante amigable para los alumnos y así trabajar de forma eficiente con el código. Se requiere un mínimo de habilidad con esta herramienta ya que debemos generar finalmente un ejecutable con el código definitivo.

**4. Proceso técnico**

**4.1 Metodología, herramientas y técnicas**

Los únicos sistemas de computación que hemos utilizado han sido nuestros ordenadores personales, y en ocasiones los ordenadores de la universidad.

Para el desarrollo del proyecto hemos establecido varias responsabilidades de acuerdo con la carga de trabajo que suponía cada tarea. Hemos establecido dos departamentos: el departamento encargado de la realización del Plan de Proyecto, y el departamento encargado de la Gestión de la Configuración del Software. Este último departamento lo hemos dividido a su vez en dos subdepartamentos, el departamento de desarrollo y el departamento de documentación, cuya coordinación entre ambos ha sido fundamental para la elaboración de este documento.

El lenguaje de programación utilizado es JAVA, y hemos utilizado la herramienta Eclipse para gestionarlo. Para poder ejecutar nuestro.jar se necesita tener en el ordenador el entorno java JRE.

Como ya hemos comentado vamos a utilizar la herramienta GitHub para subir el trabajo individual y ponerlo en común. También el Microsoft Office para definir tareas y hacer los diagramas.

También hemos complementado los conocimientos adquiridos con el paso de la asignatura y sus respectivas diapositivas con videotutoriales de Youtube cuyos links se indican en apartados anteriores y en los que nos hemos apoyado para entender mejor determinadas cosas que teníamos que realizar.

Otra herramienta fundamental que hemos utilizado ha sido la aplicación de mensajería móvil *WhatsApp* para una comunicación fluida y más rápida.

Además, hemos utilizado Skype para hacer una especie de reunión y discutir algunos temas que teníamos pendientes, esta herramienta también ha sido importante ya que muchas veces debido a la carga de trabajo que teníamos nos era imposible quedar en persona por el tiempo que teníamos que dedicarle.

Nos hemos basado en esta norma IEEE 1058.1-1987 para la *Planificación de Gestión de Proyectos Software.*

Y para el *Plan de Gestión de Configuración del Software* nos hemos basado en la norma IEEE 828-2005

**4.2 Documentación software**

El proyecto se basa en una versión base y reducida del Buscaminas que se nos da por parte del profesor, cuya especificación de requisitos es la siguiente:

* + - Se crea de forma aleatoria un tablero de juego con 30x30 casillas cuyo contenido está oculto al usuario. Cada casilla puede estar vacía, tener una mina o un número, que indica el número de minas existente en las casillas adyacentes que la rodean.
    - Se ha de utilizar el ratón para interactuar con la aplicación. El cursor permite seleccionar una casilla. Al pulsar con el botón izquierdo del ratón sobre una casilla se desvela su contenido. El botón derecho permite marcar la casilla como la posible ubicación de una mina.
    - Si se desvela una casilla vacía, se desvelarán automáticamente y de forma recursiva todas las casillas adyacentes vacías y con número. Si se desvela una mina el jugador pierde la partida, mientras que si consigue desvelar todas las casillas vacías o con número la ganará.
    - El mensaje al ganar o perder cierra el juego al pulsar aceptar, y en el caso de que la partida haya sido ganada nos mostrará además el tiempo que ha tardado el jugador.

Nuestra versión para entregar además tiene que cumplir esta especificación de requisitos adicional:

* + - Añadir opción para reiniciar el juego en cualquier momento y sin tener que reiniciar el programa.
    - Mostrar en todo momento la cantidad de minas que quedan por descubrir, así como el tiempo que llevamos jugando en la partida actual.
    - Crear diferentes niveles de dificultad:
      * Principiante (tablero 10x10 y 10 minas).
      * Intermedio (tablero 16x16 y 40 minas).
      * Experto (tablero 32x16 y 99 minas).
      * Personalizado (permite definir el tamaño del tablero y la cantidad de minas).
    - Guardar siempre los 10 mejores tiempos en cada dificultad (sin contar Personalizado).
    - En el mensaje al ganar, mostrar junto con el tiempo un campo para añadir un nombre de jugador y dar la opción de guardar el tiempo o no.
    - Permitir guardar la partida actual (estado del tablero y tiempo) en un fichero, así como poder recuperar una partida guardada a partir de un fichero.
    - El fichero donde se guarda la partida lo elegirá el usuario.
    - Crear un menú que nos permita acceder a las diferentes opciones del juego.

El código que hemos añadido al *Buscaminas.java* está debidamente comentado por el departamento encargado de la Gestión de la Configuración del Software para que este sea más intuitivo tanto para nosotros como para las personas que puedan ver ese código.

**4.3 Funciones de apoyo al proyecto**

Como apoyo al proyecto hemos utilizado la herramienta Drive para subir documentos de forma más rápida y sencilla.

También nos hemos apoyado en la aplicación de mensajería instantánea *Whatsapp* para notificar y comunicar cambios de forma rápida.

Hemos utilizado algunos vídeos de Youtube para aprender a fondo como utilizar de forma eficiente GitHub.

En ocasiones el departamento de desarrollo del código ha tenido que utilizar NetBeans en vez de Eclipse por razones de dinamismo y eficiencia, ya que Netbeans ofrece alguna funcionalidad distinta.

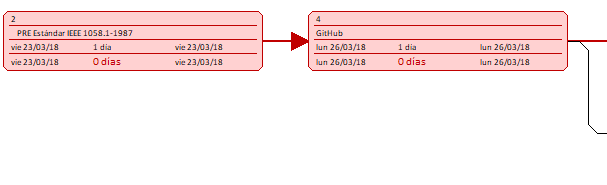
**5. Plan de desarrollo**

**5.1 Paquetes de trabajo**

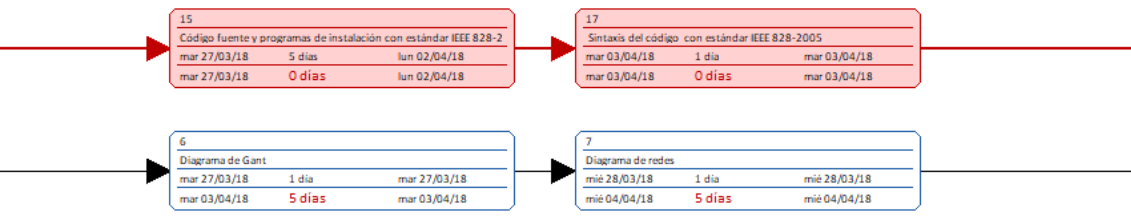
|  |  |
| --- | --- |
| Sujeto | Responsabilidades |
| Luis Herrera Collado | 1. Añadir una opción para poder reiniciar el juego en cualquier momento y sin tener que reiniciar el programa 2. Mostrar en todo momento la cantidad de minas que quedan por descubrir, así como el tiempo que llevamos jugando en la partida actual. 3. Crear el nivel de dificultad Principiante 4. Crear el nivel de dificultad Intermedio 5. Crear el nivel de dificultad Experto (tablero 32x16 y 99 minas) 6. Crear el nivel de dificultad Personalizado (permite definir el tamaño del tablero y la cantidad de minas). 7. Guardar los 10 mejores tiempos en cada dificultad (sin contar Personalizado) 8. En el mensaje al ganar, mostrar junto con el tiempo un campo para añadir un nombre de jugador 9. Dar la opción de guardar el tiempo o no. 10. Permitir guardar la partida actual (estado del tablero y tiempo) en un fichero, así como poder recuperar una partida guardada a partir de un fichero. 11. Crear un menú que nos permita acceder a las diferentes opciones del juego 12. Ser el principal gestor de Github y gestionar las modificaciones de los otros participantes del proyecto al interactuar con la plataforma. 13. También debe colaborar con el responsable del  Plan de Gestión de Configuración del Software según especificaciones del estándar IEEE 828-2005 14. Es responsable de repaso y control del proyecto. |
| Miguel Sánchez Pérez | 1)Gestión de riesgos          2)Mecanismos de supervisión y control           3) Plan de persona           4) Metodología, técnicas y herramientas           5)Documentación software           6)Funciones de apoyo al proyecto            7)Paquetes de trabajo            8)Dependencias,            9)Recursos           10)Presupuesto           11)Distribución de recursos           12) Calendario,           13)Secciones adicionales           14)Índice           15)Apéndices.           16) Colaboraracción en el diagrama Gant           17) Colaboraracción diagrama Pert.           18) Colaboraracción  diagrama Red.           19) Suposiciones, dependencias y restricciones           20)Reponsable de repaso y control del proyecto. |
| Guillermo Roldán Peigneux d’ Egmont | Responsable del PGCS:       1) Introducción                 a) Objetivo                  c)Propósito                  d)Alcance                   e)Definiciones y acrónimos                   f) Referencias        2) Gestión de la GCS                    a) Objetivo                    b) Organización                    c) Responsabilidades de la GCS                    d) Políticas, directivas y procedimientos aplicables        3) Actividades de GCS                       a) Descripción de las tareas de GCS                       b) Control de la interfaz                       c) Control del “subcontratista/vendedor         4)Calendarios de GCS         5)Recursos de GCS         6) Mantenimiento del plan de GCS          7)Colaboracción con la realización del código          8) Repaso y control del proyecto. |
| Jorge Barrio Castro | Responsable de:  1)Visión General del proyecto .  2)Productos finales .  3)Evolución del Plan de Proyecto .  4)Documentos de referencia .  5)Definiciones y acrónimos  6) Modelo de procesos             a)Realización del Diagrama de Gant             b) Realización del Diagrama de Pert             c) Realización del Diagrama de Redes  7)Estructura organizativa  8) Fronteras e interfaces organizativa  9)Responsabilidades,  10)Objetivos y prioridades de gestión.  11) Reponsable de repaso y control del proyecto. |

**5.2 Dependencias**

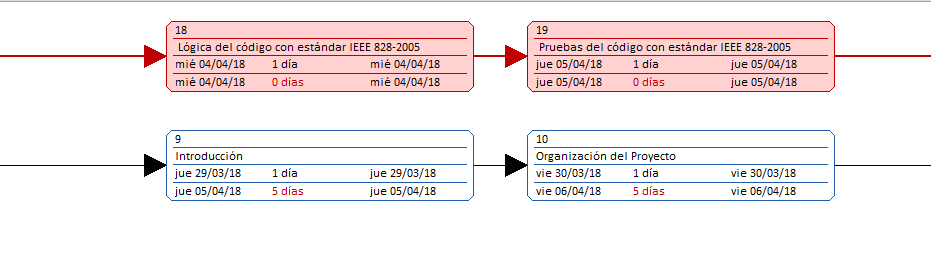
Para la descripción de este punto hemos utilizado el diagrama de redes que nos proporciona Microsoft Project. Aquí se muestran claramente las interdependencias entre actividades. Hemos cambiado el formato de tarjeta para crear nuestro propio diseño. Las tareas que están en paralelo fueron realizadas a la vez pero por distintos departamentos.



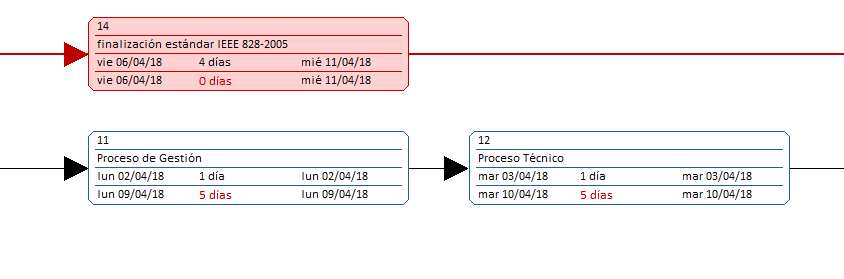
*Diagrama de red - Parte 1*



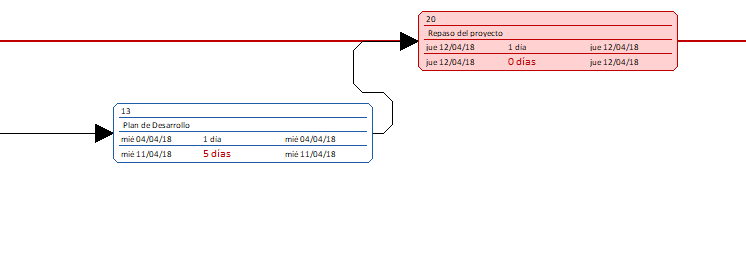
*Diagrama de red: Parte 2*



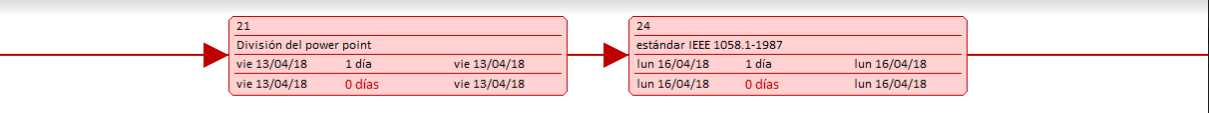
*Diagrama de red: Parte 3*



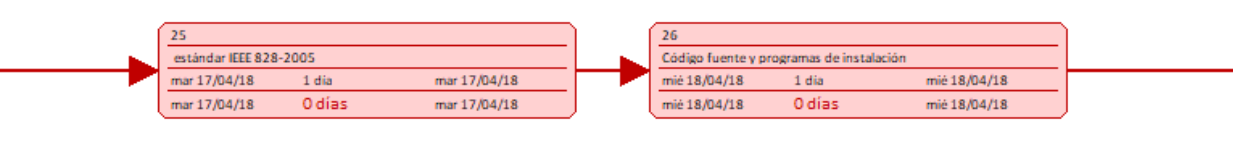
*Diagrama de red: Parte 4*



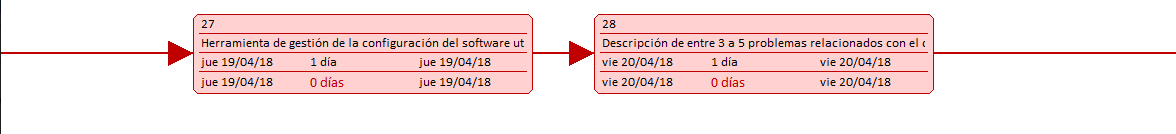
*Diagrama de red: Parte 5*



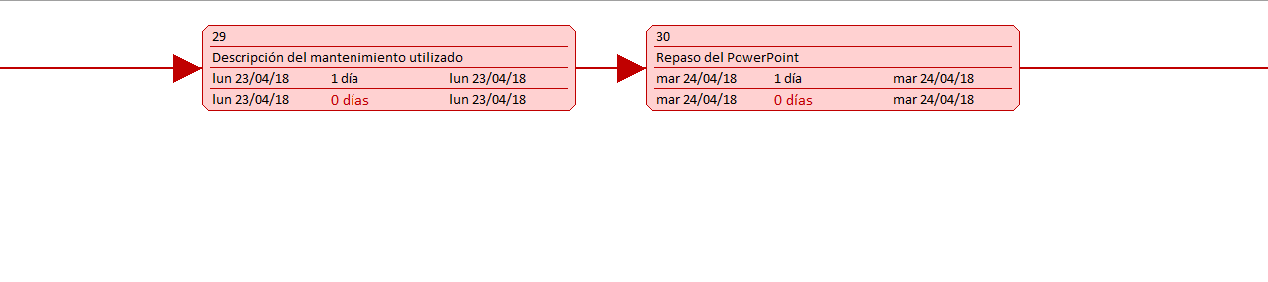
*Diagrama de red: Parte 6*



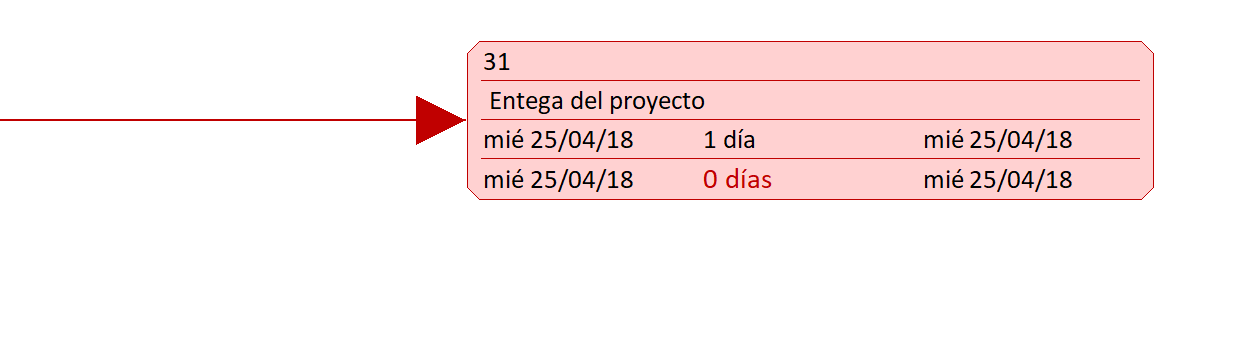
*Diagrama de red: Parte 7*



*Diagrama de red: Parte 8*



*Diagrama de red: Parte 9*



*Diagrama de red: Parte 10*

**5.3 Recursos**

Somos cuatro estudiantes de la Ingeniería Informática los que realizamos este proyecto y hemos utilizado distintos tipos de recursos para completarlo.

Hemos trabajado principalmente con ordenadores donde hemos desarrollado el código nuevo que se nos pedía mediante Eclipse en este caso.

A su vez, hemos usado el Word como herramienta principal para generar los informes pertinentes de cada parte.

El Microsoft Project, recomendado por el profesor, nos ha sido de gran ayuda para generar los diagramas de Pert y de Gant, los cuáles sirven posteriormente para completar el PGPS. También para hacernos un diagrama relacionando las tareas y el tiempo invertido en cada una.

GitHub nos ha permitido subir cada uno nuestro trabajo a la plataforma y así hacerlo común.

También mediante el Eclipse hemos generado el .jar que es el ejecutable que entregaremos a los profesores para que sea evaluado.

**5.4 Presupuesto y distribución de recursos**

Todas las herramientas y recursos que hemos necesitado han sido gratuitos y en esta práctica no era necesario ningún presupuesto ya que estamos realizando un proyecto académico.

A su vez, todas las herramientas las hemos obtenido o bien de internet, como es el caso de GitHub, Microsoft Project y Eclipse, o porque ya los teníamos previamente instalados en nuestros ordenadores como el Word.

En las ocasiones que hemos hecho uso de los ordenadores del aula de la universidad, las herramientas necesarias estaban subidas en myApps, por lo que no nos ha supuesto ningún problema trabajar con los ordenadores de la universidad

**5.5 Calendario**

