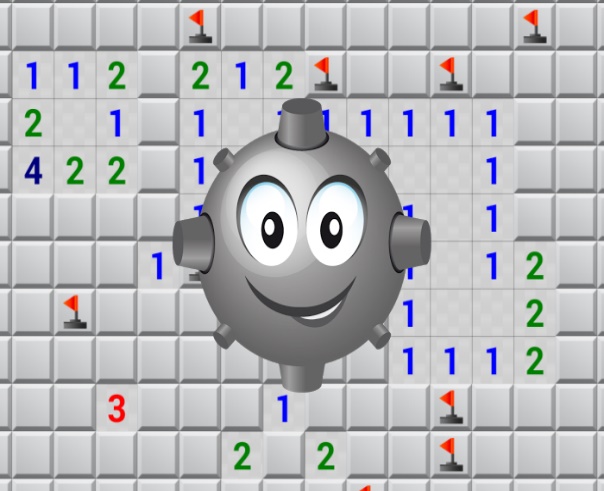


PLAN DE GESTIÓN DE PROYECTO



25/04/2018

"URJC"

Buscaminas

PLAN DE GESTIÓN DE PROYECTO

# Tabla de contenidos

[**Prefacio……………………………………………………………………………………………………………………………….………………….1**](#_Toc511387136)

[**1.Introducción……………………………..………………………………………………………………………………………………………….1**](#_Toc511387137)

[1.1Visión General del proyecto………………………………………………………………………...…………………………………1](#_Toc511387138)

1.2 Productos finales ……………………………………………………………………………………………………………………….1

1.3 Evolución del Plan de Proyecto ………………………………………………………………………………….…………………1

1.4 Documentos de referencia…………………………………………………………………………………………………………….1

1.5 Definiciones y acrónimos………………………………………………………………………………………………..…………….1

**2.Organización del Proyecto………….…………………………………………………………………………………………..……………1**

2.1 Modelo de procesos………..…………………………………………………………………………………………………………....1

2.2 Estructura organizativa……………..…………………………………………………………………………………………………1

2.3 Fronteras e interfaces…..………………………………………………………………………………………………………………1

2.4 Responsabilidades……………………………………..………………………………………………………………………………...1

**3.Proceso de Gestión…………………………………………………………………………………………………………………..…………..1**

3.1 Objetivos y prioridades de gestión………………………………………………………………………………………………...1

3.2 Suposiciones, dependencias y restricciones…………………..………………………………………………………………1

3.3 Gestión de riesgos………………………………………………………………………………………………………………………...1

3.4 Mecanismos de supervisión y control……………………………………………………………………………………………1

3.5 Plan de personal…………………………………………………………………………………………………………………………..1

**4.Proceso Técnico……………………………………………………………………………………………………………………………………1**

4.1 Metodología, técnicas y herramientas……………………………………………………………………………………………1

4.2 Documentación software………………………………………………………………………………………………………………1

4.3 Funciones de apoyo al proyecto…………………………………………………………………………………………………….1

**5.Plan de desarrollo……………………………………………………………………………………………………..…………………………1**

5.1 Paquetes de trabajo……………………………………………………………………………………………………………………...1

5.2 Dependencias………………………………………………………………………………………………………………………………1

5.3 Recursos………………………………………………………………………………………………………………………………………1

5.4 Presupuesto…………………………………………………………………………………………………………………………………1

5.5 Calendario……………………………………………………………………………………………………………………………………1

**Índice………………………………………………………………………………………………………………………………………………………1**

**Apéndices………………………………………………………………………………………………………………………………………………..1**

# Prefacio

Este informe, hace referencia al conjunto de acciones necesarias para alcanzar el los objetivos y necesidades establecidas en el enunciado de la practica obligatoria. Estos elementos son:

* Mejoras las funcionalidades del programa “Buscaminas”
  + Selección del nivel de dificultad que solicite el usuario o configuración de su propio nivel.
  + Implantación de un menú superior que contenga información de distinta índole. Podemos destacar las siguientes:
    - Contador de minas marcadas.
    - Tiempo del juego.
    - Opción de reiniciar la partida.
    - Opción de crear una nueva partida indicando consigo el nivel en el que se quiere jugar.
    - Guardar la partida antes de su finalización.
    - Tabla de los mejores tiempos de cada nivel con el nombre del jugador que haya alcanzado dicho tiempo record.
    - Archivar el tiempo la partida y el nombre del jugador en el caso de que se haya superado el tiempo record del nivel jugado.
    - Almacenamiento en un fichero de texto de los mejores tiempos en cada uno de los niveles, así mismo, el nombre del jugador.
    - Almacenamiento en un fichero de texto de una partida sin terminar que quiera guardar el usuario.
* Plan de gestión de configuración de software tiene como objetivo maximizar la producción minimizando los errores aplicado a cada fase del proceso de software. Por ello, este plan trata de gestionar la los cambios a lo largo de toda la vida del software.
* Plan de mantenimiento, hace referencia al conjunto de cambios realizados en un software determinado después de su implantación.

# Introducción

## VISIÓN GENERAL DEL PROYECTO

Como se ha indicado en el apartado anterior, el objetivo de este trabajo consiste en mejorar las funcionalidades del juego *Buscaminas* a partir de un código ya proporcionado. Para conseguir estas mejoras seguiremos los siguientes pasos:

1. Verificación del código proporcionado: realizaremos una lectura activa y probaremos el funcionamiento del mismo para comprender mejor su funcionalidad.
2. Validación del código: cada uno de los integrantes dará el visto bueno en relación a la comprobación de los elementos del *Buscaminas* básico proporcionado.
3. Realización de las mejoras
4. Comprobación de errores y mejora del código implementado.
5. Puesta en común y acciones de mejora: en el momento en el que tengamos todas las funcionalidades requeridas en el proyecto, todos los integrantes del grupo nos reuniremos para comprobar el correcto funcionamiento de las acciones integradas y posibles ideas para mejorar su implantación dentro del código.

Como acabamos de indicar, este proyecto se realizará siguiendo un proceso ordenado y lógico de acción para alcanzar los requisitos y las funcionalidades requeridas por el cliente. Una vez finalizada la construcción de la arquitectura del juego, pasaremos a la fase de mantenimiento en la que iremos resolviendo los problemas que surjan tanto a corto plazo como a largo.

Para poder simplificar el proceso, utilizaremos la plataforma GitHub que nos permitirá tener un repositorio común en el que podemos subir todas las modificaciones que hagan los integrantes de nuestro grupo además de poder trabajar de forma remota.

Por otro lado, utilizaremos la aplicación *NetBeans* para programar las funcionalidades que se nos solicitan e ir probando los cambios que se hagan en el código.

## PRODUCTOS FINALES

Como productos finales podemos identificar, en primer lugar, el Plan de Proyecto que se trata del informe inicial que indica los elementos iniciales, la forma de implementarlos y el método que seguiremos para alcanzar los requisitos indicados por el cliente.

En segundo lugar, el Plan de Gestión de Configuración del Software con el que conseguimos mantener la integridad de los elementos y garantizar que no se realizan cambios no controlados y que todos los participantes del proyecto disponen de la versión adecuada de los productos que manejan. De este modo, podemos demostrar al cliente el proceso que hemos seguido y la forma en la que hemos implementado los cambios.

En tercer lugar, el programa *Buscaminas* con las mejoras solicitadas por el cliente definidas en los puntos anteriores.

En cuarto lugar, el Plan de Mantenimiento realizado una vez entregado el proyecto final al cliente y realizado con el fin de establecer las prácticas, recursos y secuencias de actividades relevantes para mantener el producto.

Por último, la presentación PowerPoint que se usará un día concreto para poder exponer el trabajo.

Su fecha de entrega corresponde al 25 de abril de 2018.

## EVOLUCIÓN DEL PLAN DE PROYECTO

El plan de proyecto se llevara a cabo por el conjunto del grupo, cada uno se encargará de uno de los puntos de este informe. Para que uno de los integrantes pase a hacer su apartado, primero se evaluará y aprobará entre todo el grupo las modificaciones sobre este plan, por lo tanto, se modificará cada vez que se haya aceptado los cambios. Todos estos cambios serán comunicados a través de dos canales distintivos:

* Puesta en común sobre los cambios realizados.
* Plataforma *GitHub* en la que podemos ir subiendo las actualizaciones tanto de los informes como del código.

Este Plan deberá ser revisado al inicio de cada fase, modificado de acuerdo a lo necesario, aprobado y distribuido al equipo de proyecto.

## DOCUMENTOS DE REFERENCIA

Los documentos sobre los que nos basaremos en el desarrollo de los informes son, principalmente, los siguientes:

* Norma IEEE 1058.1-1977 para la planificación de gestión de proyectos software
  + Número de informe:
  + Autor:
  + Organización que lo ha publicado:
* Gestión y dirección de proyectos software
  + Autor: Sin identificar
  + Organización que lo ha publicado: Universidad Rey Juan Carlos en el aula virtual

## DEFINICIONES Y ACRÓNIMOS

Se tiene que ir añadiendo según se vaya usando.

# Organización del proyecto

## MODELO DE PROCESOS

Se van a establecer una serie de relaciones, a lo largo del conjunto del proyecto, entre las actividades principales del proceso. Para conseguir este propósito, dividiremos en una serie de tareas el conjunto del proyecto con el fin de poder establecer estas relaciones, ya que no se puede empezar una actividad hasta que no haya finalizado otra.

De este modo, algunas de las actividades que identificaremos serán dependientes de otras, es decir, no se puede empezar otra actividad hasta que no se haya finalizado la otra, ya que una depende de la otra. Haciendo referencia a nuestro proyecto, la actividad de validación del código no puede empezar hasta que no se haya terminado de verificar el código.

Por otra parte, las 3 actividades principales del proyecto, a su vez se dividen en más tareas para tener una mejor percepción del tiempo que se dedicará a cada actividad.

De esta forma, el proyecto se divide en:

1. **Verificación de código** ( dependencias: 2 )

Tiempo total: 1 día

1. **Validación de código** ( dependencias: 3 )

Tiempo total: 1 día

1. **Versión actualizada** (dependencias: 4 )

Tiempo total: 15 días

* 1. Añadir opción para reiniciar el juego 🡪 1 día
  2. Mostrar cantidad de minas 🡪 1 día
  3. Mostrar tiempo de partida 🡪 1 día
  4. Niveles de dificultad 🡪 3,2 días
  5. Guardar 10 mejores tiempos de cada dificultad 🡪 1 día
  6. Mostrar 10 mejores tiempos de cada dificultad 🡪 1 día
  7. Añadir un nombre de jugador 🡪 1 día
  8. Dar la opción de guardar el tiempo o no 🡪 1 día
  9. Permitir guardar la partida actual en un fuchero 🡪 3,2 días
  10. Permitir consultar la última partida jugada 🡪 1 día
  11. Recuperar una partida guardada de un fichero 🡪 1 día
  12. Menú para acceder a las diferentes opciones del juego 🡪 1 día

1. **Validación de código actualizado** ( dependencias: ninguno )

Tiempo total: 1 día

* 1. Validación del código actualizado 🡪 1 dia
  2. Pruebas del código 🡪 1 día

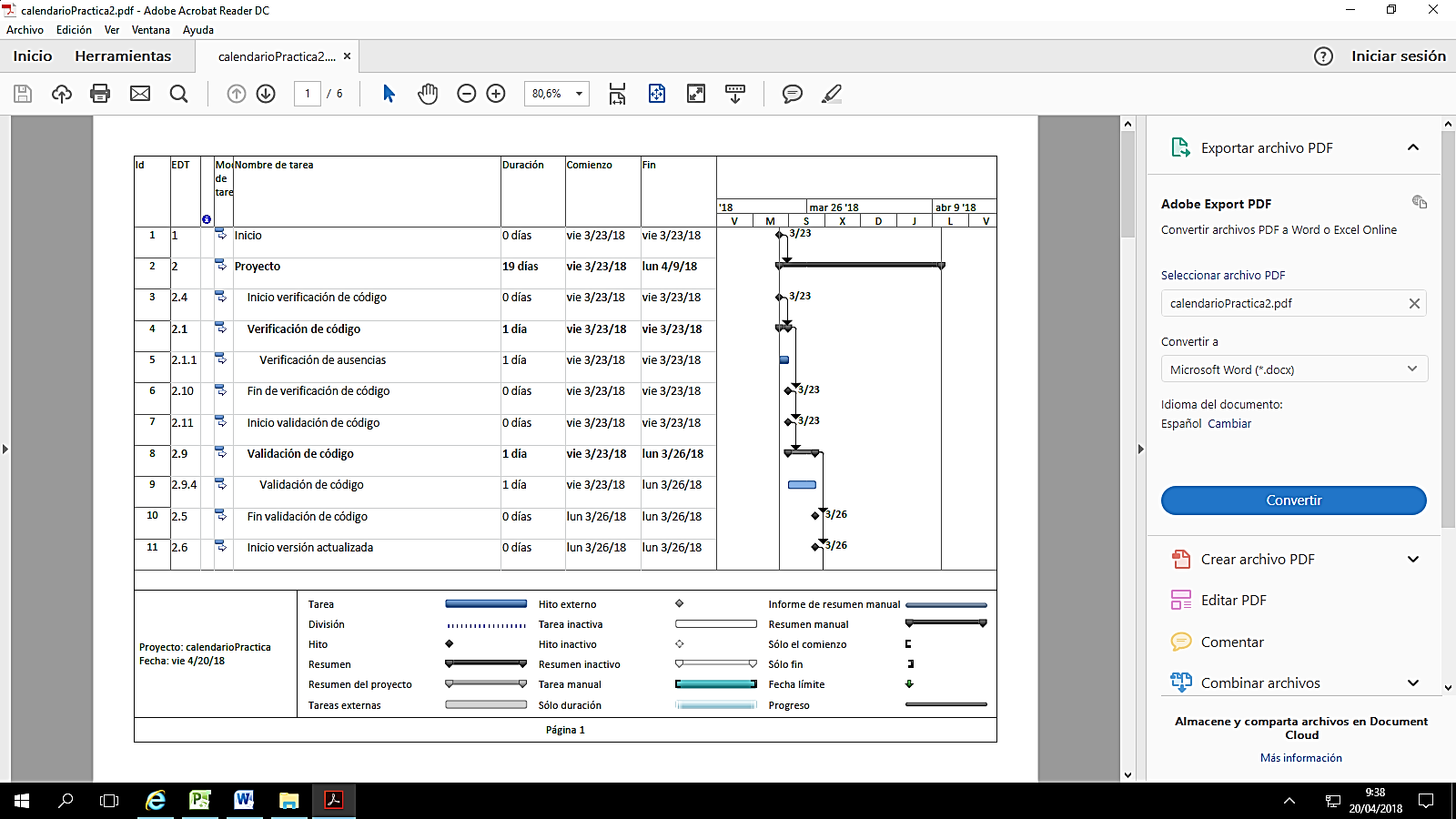
1. **Puesta en común del proyecto** ( dependencias: 3,4 )

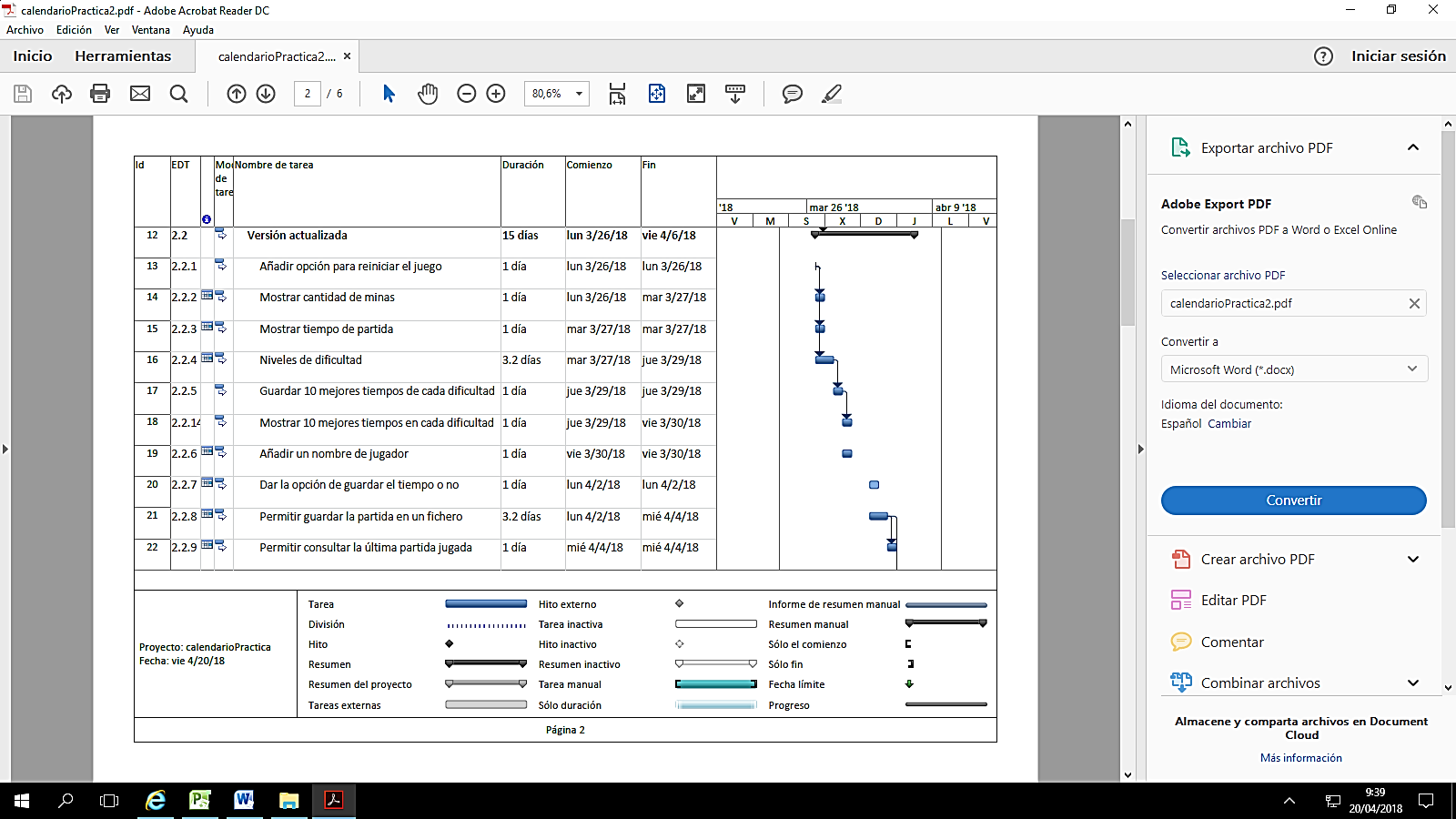
Tiempo total: 1 día

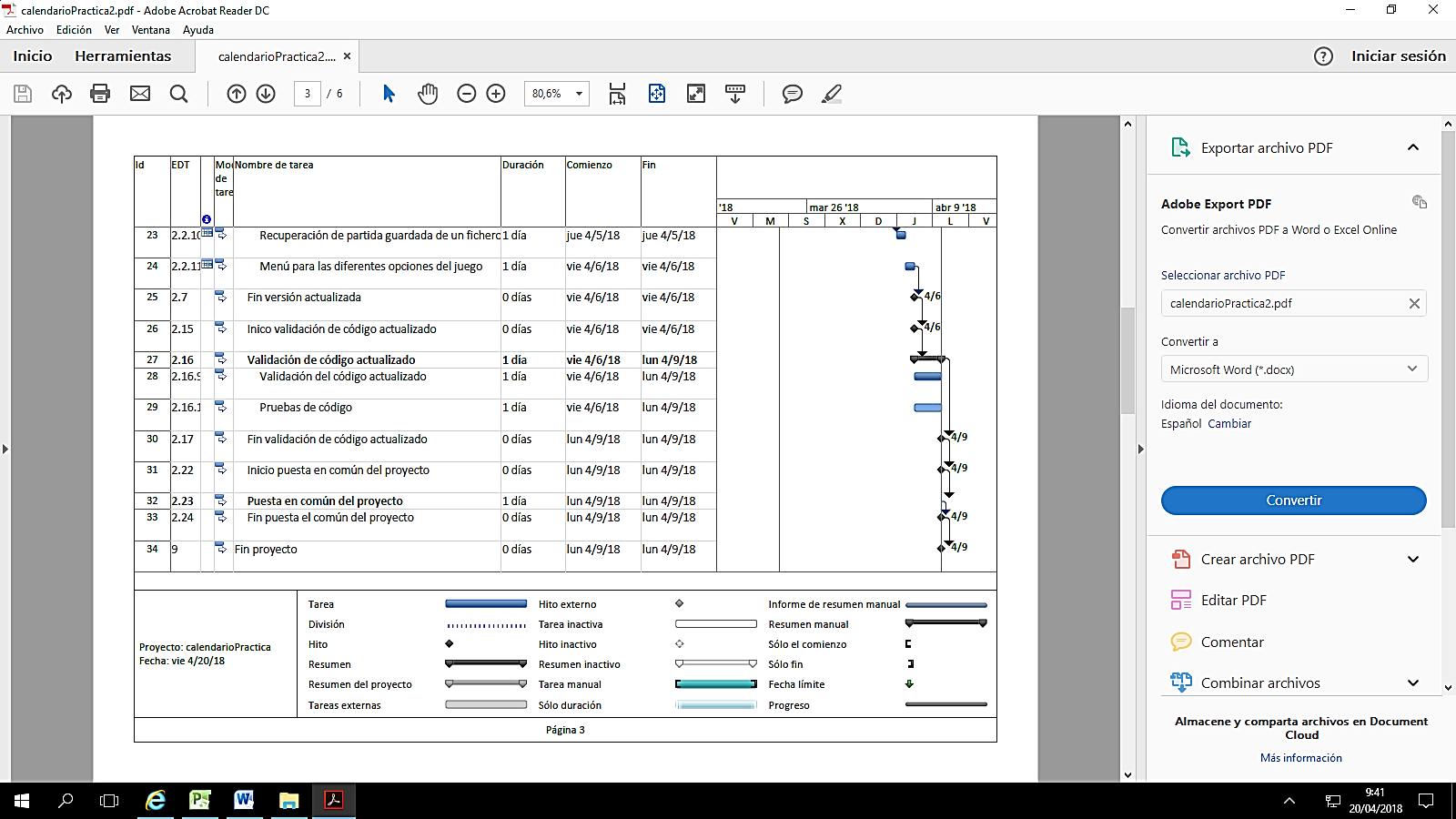
La suma total del tiempo es de: 19 días. (sin tener en cuenta que el número de personas dedicadas al proyecto son 4 y que las actividades se podrán realizar el paralelo)

## ESTRUCTURA ORGANIZATIVA

Para la gestión interna del proyecto se ha usado el diagrama de Gantt con el fin de poder ver la organización de las actividades, ya que las diferentes actividades se llevan a cabo entre uno o varios miembros del grupo.







Las actividades a realizar por cada miembro del grupo son:

* Carlos: 2.1, 2.9, 2.2.7, 2.2.6, 2.2.9, 2.2.10, 2.16, 2.23
* Jorge: 2.2.1, 2.2.2, 2.2.3, 2.2.8, 2.2.9, 2.2.10, 2.16, 2.23
* Alvaro: 2.2.5, 2.2.7, 2.2.14, 2.2.9, 2.2.10, 2.16, 2.23
* Adina: 2.2.4, 2.2.9, 2.2.11, 2.16, 2.23

## FRONTERAS E INTERFACES ORGANIZATIVAS

La organización encargada del proyecto es un equipo formado por 4 personas. Todo el proyecto se ha hecho a nuestro cargo, siendo la organización cliente todas aquellas personas que utilicen el juego, independientemente de la plataforma (*Windows, Linux, Mac...)* que tengan en sus terminales. No contamos con ninguna organización subcontrada.

Se ha verificado, por los integrantes del proyecto que el juego funciona adecuandamente para poder cumplir con los requisitos de calidad del producto.

## RESPONSABILIDADES

(METER MÁS AQUI SI NO LLEGAMOS A 30 HOJAS)

Las responsabilidades de cada miembro han sido expuestos en el punto 2.2. El Plan de Proyecto y el plan de Gestión de Configuración del Software han sido responsabilidad de todos los miembros del equipo, logrando así hacer el trabajo más ameno e incluso favoreciendo el trabajo en grupo a la hora de generar ideas para aquellos apartados en los que no se sabía que añadir.

Por lo tanto, podemos resumir el proyecto en 3 grandes actividades:

1. Juego de buscaminas
   1. Código Java con sus correspondientes mejoras
2. Presentación : Power Point
3. Documentos
   1. Plan de proyecto
   2. Plan de Gestión de Configuración

Actividades de las cuales, todos los miembros del equipo han sido responsables.

# Procesos de gestión

## OBJETIVOS Y PRIORIDADES DE GESTIÓN

La prioridad absoluta es hacer que el juego funcione correctamente, sobre todo que guarde las partidas generadas en un fichero para poder visualizar en cualquier momento, los 10 mejores tiempos. Con esto, se conseguirá tener en un archivo todas las partidas jugadas.

(WTF’??)

## SUPOSICIONES, DEPENDENCIAS Y RESTRICCIONES

El juego buscaminas tiene que tener un funcionamiento similar al juego original. Por lo tanto, como restricciones hemos añadido que el usuario tiene que elegir como entrada uno de los niveles de dificultad proporcionados (principiante, intermedio, experto o personalizado). No se pone el juego predeterminado.

Por otro lado, en cuanto a las opciones que facilita el menú, *reiniciar* siempre carga el mismo nivel que el usuario ha elegido anteriormente y *juego* *nuevo* da la opción de volver a elegir un nuevo nivel, es decir, volvería a la pantalla de inicio del juego.

En cuanto al nombre del usuario como restricción se ha añadido que vaya sin espacio, para poder buscar luego en los ficheros ya que se han usado los espacios como separador de búsqueda. Si el usuario metiese espacios en el nombre con el que guardará la partida, saldrá una pop up informando al usuario de la restricción correspondiente. Además, si el usuario no supera un record de tiempo de ese nivel, no se le da la opción de guardar el nombre.

## GESTIÓN DE RIESGOS

Para evitar la pérdida de información, se ha decido utilizar la plataforma de GitHub, a través de la cual todos los miembros del grupo podrán ver en cada momento quién está realizando algún cambio en la información así como en qué parte del código se realiza ese cambio.

Sin embargo, exponerlo a una plataforma pública como es GitHub suponía el riesgo de que otros grupos que realizasen el mismo proyecto se aprovechase de la información proporcionada y copiase de esta forma partes del código que este grupo ha hecho.

Por lo tanto, para poder gestionar ese riesgo, el código se ha ido haciendo en el mismo terminal y sólo se ha subido a la plataforma cuando los integrantes del grupo no se han podido reunir y han tenido que trabajar de forma paralela desde sus respectivos establecimientos o lugares de trabajo.

Por otro lado, se ha guardado una copia de seguridad, después de realizar cada cambio en el ordenador de todos los miembros del grupo, copia que se enviaba por correo electrónico a todos los miembros.

## MECANISMOS DE SUPERVISIÓN Y CONTROL

Por cada tarea elaborada, como por ejemplo, añadir los niveles del dificultad al juego o mostrar el tiempo, se ha ido ejecutando el código a través de unas pruebas para comprobar que todo funcionaba perfectamente. Estas pruebas hay que tener en cuenta que se realizan siguiendo las restricciones descritas anteriormente, ya que sino, el juego no funcionaría de forma adecuada.

Para comprobar que las restricciones se aplicaban correctamente, se han ido metiendo nombres con espacios pudiendo así, ver si salía la pop up correspondiente.

Además, en cada nivel de dificultad, después de realizar la tarea que facilitase la elección de dicho nivel, se ha comprobado que el tablero del juego salía tal y cómo indicaban las restricciones. Hay que añadir que en el nivel de personalizado, se ha hecho un control más exhaustivo para comprobar que el tablero se creaba con las medidas especificadas por el usuario.

Por otro lado, una vez añadida la tarea de guardar los 10 mejores tiempos, todos los miembros del grupo jugaron durante media tarde al juego, controlando así la inserción de las partidas dentro del fichero, con el correspondiente nombre de usuario y su tiempo. Gracias a este control, se ha podido comprobar que dentro de los ficheros, los tiempos se organizan de forma ascendente en función del tiempo de la partida, desde el mejor tiempo hasta el peor.

El mismo control se ha llevado a cabo después de realizar la tarea de guardar las partidas en un fichero. Se ha comprobado que el fichero tenía la partida actual correspondiente, así como que se podía ver la última partida jugada. Así como que se podía recuperar del fichero la última partida jugada. Comprobamos esto último incluso con unos días de por medio para ver si podía recuperar la partida aunque no fuese en el mismo día.

## PLAN DE PERSONAL

(meter PERT)

# Proceso Técnico

## METODOLOGÍA, TÉCNICAS Y HERRAMIENTAS

## DOCUMENTACIÓN SOFTWARE

Se ha utilizado para la realización de los distintos apartados de este documento, los siguientes programas:

* Diagrama de Gantt: se ha utilizado el programa de Microsoft Project 2010
* Tabla de Precedencias: se ha utilizado Microsoft Excel 2016
* Plan de proyecto: se usará el Microsoft Word 2016
* Plan de mantenimiento: se usará el Microsoft Word 2016
* Plan de Gestión de Configuración: se usará el Microsoft Word 2016
* Diagrama PERT: se ha utilizado un programa llamado inkscape, versión 0.92.3
* Buscaminas: programado en NetBeans usando el lenguaje Java

Además se presenta también un Plan de Gestión de Configuración que sigue el estándar IEEE 282-2005 en el que se especifican todas las actividades de Gestión de Configuración y Cambios que serán realizados durante todo el ciclo de vida del proyecto.

## FUNCIONES DE APOYO AL PROYECTO

# Plan de desarrollo

## PAQUETES DE TRABAJO

## DEPENDENCIAS

## RECURSOS

## PRESUSPUESTO Y DISTRIBUCIÓN DE RECURSOS

## CALENDARIO

# Índice

# Apéndices