

2019  
2020

# Gestión de la configuración del software

Ampliación de Ingeniería del software

GRUPO 8

AGUDO RIVAS, DANIEL  
CORTES MORALES, NOELIA  
DE PERINAT GOMEZ, OSCAR  
LORENZO ALONSO, DANIEL

## Contenido

1.0 - Introducción a la GCS .....	2
1.1 – Ámbito de aplicación a nuestro proyecto .....	2
2.0 - Gestión de la configuración del software.....	2
2.1 - Actividades de la configuración del SW .....	2
2.2 – Control de la configuración del SW .....	3
2.3 - Planificación del proyecto y Calendario .....	4
2.4 - Contabilidad de la configuración del software .....	5
3.0 – Recursos.....	6
4.0 – Mantenimiento .....	6

## 1.0 - Introducción a la GCS

En este documento vamos a abordar la **Gestión de la Configuración del Software** (en adelante GCS) recordando previamente su definición.

La GCS se define como *disciplina que identifica la configuración de un sistema en puntos discretos del tiempo, con el objetivo de controlar sistemáticamente los cambios de esa configuración y mantener su calidad y trazabilidad a través del ciclo de vida del sistema.*

La finalidad de este proceso es la de controlar todas las posibles modificaciones que se puedan realizar en un proceso de desarrollo software (en adelante SW) minimizando así los posibles riesgos y errores que puedan surgir durante el proceso de desarrollo de SW.

### 1.1 – Ámbito de aplicación a nuestro proyecto

El desarrollo de este documento tiene como propósito establecer las actividades relativas a la GCS que hemos realizado en el proyecto **TicTacToe**. En él contemplamos el trabajo realizado por cada integrante del grupo tanto a nivel de programación como en la realización de toda la documentación requerida durante el proceso de desarrollo.

## 2.0 - Gestión de la configuración del software

### 2.1 - Actividades de la configuración del SW

La función de esta actividad es registrar los elementos que vamos a controlar asignando identificadores únicos para los distintos elementos y versiones del proyecto. La finalidad es llevar un control organizado de los elementos de datos del proyecto SW.

#### Nombrado de los elementos

Identificador único	XXX-YYY-Z-RL-NNN
XXX	Identificador de la empresa
YYY	Identificador del proyecto TicTacToe
<b>Z</b>	<b>Identificador del tipo siguiente</b>
P	Plan
R	Especificación de requisitos
D	Diseño
S	Listado fuente
T	Pruebas
U	Manual de usuario
I	Guía de instalación
M	Mantenimiento
RL	Nivel de revisión
NNN	Código de atributo, definido por los desarrolladores de SW. Refleja información importante de los elementos de la configuración.

## Aplicación al proyecto

Identificador único	XXX-YYY-Z-RL-NNN
URJC-TTT-P-00-0220	Plan del proyecto asignado TicTacToe (TTT) de la empresa Universidad Rey Juan Carlos (URJC). Documento original (00) dispuesto al control de cambios en Marzo de 2020(0220).
URJC-TTT-R-01-0220	Primera revisión (R-01) de la especificación de requisitos del proyecto (TTT). Dispuesto al control de cambios en marzo de 2020 (0220).
URJC-TTT-D-02-0220	Segunda revisión del diseño referente al EC Player. Dispuesto al control de cambios en febrero de 2020
URJC-TTT-D-03-0320	Tercera revisión del diseño referente al StatisticsService. Dispuesto al control de cambios en marzo de 2020.
URJC-TTT-D-04-0320	Cuarta revisión del diseño referente al TicTacToeHandler. Dispuesto al control de cambios en marzo de 2020
URJC-TTT-D-05-0320	Quinta revisión del diseño referente al statisticsController. Dispuesto al control de cambios en marzo de 2020.
URJC-TTT-D-06-0320	Sexta revisión del diseño referente al Templates y Mustache. Dispuesto al control de cambios en marzo de 2020.

### 2.2 – Control de la configuración del SW

Se identifica la necesidad de realizar cambios en el proyecto, viabilidad y aprobación de los mismos. Todo ello debidamente documentado.

Nombre y versión de los EC	Nombre y organización del autor de la solicitud	Fecha de petición	Necesidad del cambio	Descripción del cambio
Player v1	Grupo 8	25 Febrero 2020	Recogida y almacenamiento de datos	Añadimos atributos y métodos para recoger los datos de los jugadores y manipularlos
StatisticsService v1	Grupo 8	25 Febrero 2020	Almacenamiento de datos	Creación de métodos para actualizar todos los datos referentes al jugador
TicTacToeHandler v1	Grupo 8	25 Febrero 2020	Captura de datos en los diferentes eventos que se producen	Actualización de los diferentes estados por los que pasa el juego. Modificamos el método handleMessage()
StatisticsController v1	Grupo 8	25 Febrero 2020	Recogida de estadísticas para generar la página HTML	Creación de la vista y referencia para mostrarla.
StatsMustache v1	Grupo 8	25 Febrero 2020	Visualización de las estadísticas de los jugadores	Creación de la página .html para mostrar las estadísticas recopiladas y mostrarlas.

### Implementación de los cambios en el proyecto

Tras examinar los requisitos del cliente, y, para adaptar los requerimientos del mismo, planteamos en el proyecto una serie de cambios en las versiones del software que

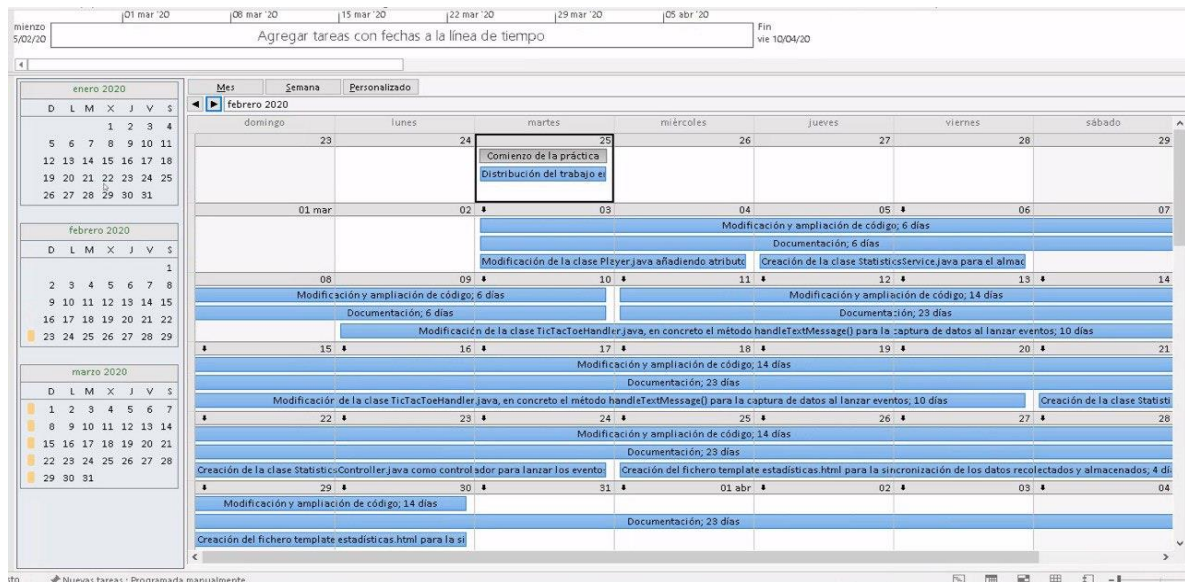
desarrollamos así como la inclusión de nuevas clases y métodos. Los cambios se han consensado en el equipo y se han decidido llevar a cabo como se muestran en la tabla así como las fechas y versiones en los que han derivado.

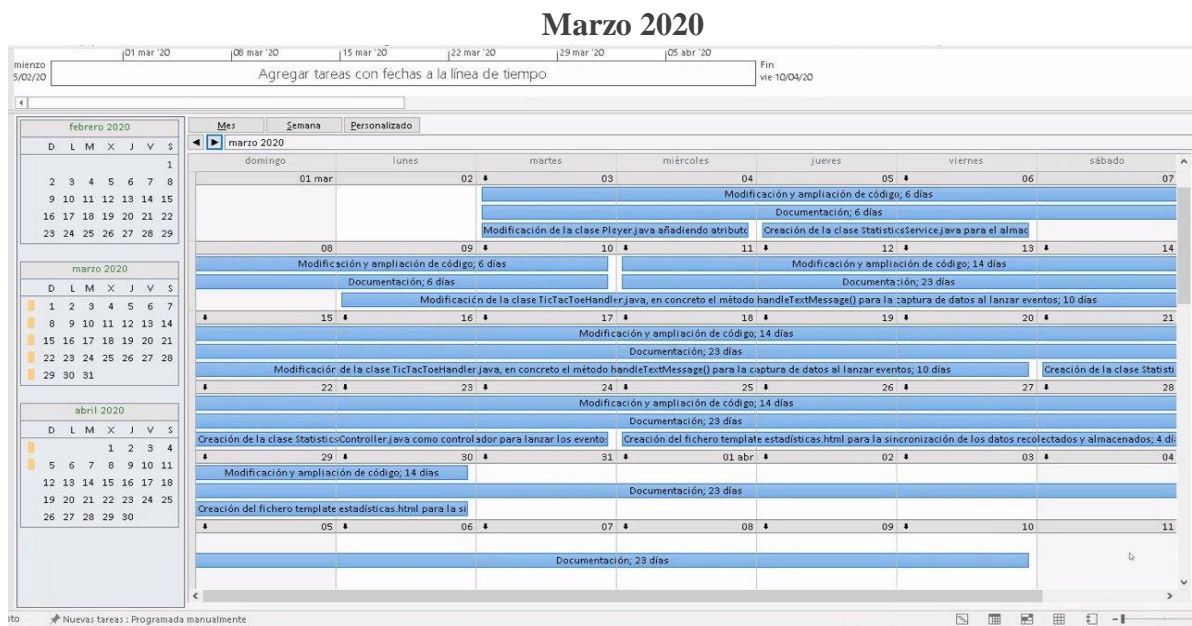
Petición de cambio asociada	Nombres y versiones de los ECs afectados	Fechas de liberación	Identificador de la nueva versión
Player.java	Player v.1	03 Marzo 2020	Player v.2
statisticsService.java	Nueva Clase añadida	5 Febrero 2020	StatisticsService v.1
TicTacToeHandler.java	TicTacToeHandler v1	09 Marzo 2020	TicTacToeHandler v.2
TicTacToeHandler.java	TicTacToeHandler v.2	15 Marzo 2020	TicTacToeHandler v.3
StatisticController.java	Nueva clase añadida	22 Marzo 2020	StatisticController v.1
Creación Templates estadísticas	Nuevo recurso añadido	25 Marzo 2020	Estadísticas.html

### 2.3 - Planificación del proyecto y Calendario

En esta sección adjuntamos una serie de ficheros donde indicamos el tiempo que nos ha llevado realizar las diferentes actividades que hemos realizado durante el mes de febrero y el mes de marzo.

#### Febrero 2020





## 2.4 - Contabilidad de la configuración del software

Los cambios realizados se han ido realizando de forma secuencial, en ocasiones en paralelo entre varios de los recursos que se han visto implicados. Al poder realizar cada miembro del grupo una parte de la programación se ha hecho de una forma más viable que si el control de cambios hubiese sido realizado solo por una parte del mismo por lo que a esto respecta hemos tenido un ahorro significativo del tiempo empleado en el desarrollo.

Hemos modificado la clase `Plajer.java` con la inclusión de nuevos atributos que nos permitan llevar un control de las partidas de cada jugador llevando a la nueva versión de software que se ha indicado con anterioridad.

La clase `TicTacToeHandler.java` nos ha llevado más tiempo el poderla realizarla ya que hemos adecuado las estructuras de datos que conocíamos a la implementación realizada así como trazar todo el software desarrollado hasta llegar a identificar que las estadísticas las estábamos recogiendo de la forma que se exige para satisfacer los requisitos del cliente.

En la clase `StatisticsService.java` hemos recogido toda la estructura en memoria que nos va a gestionar los datos para cada jugador, en definitiva, es la encargada de actualizar todos los datos de cada jugador.

Las clases `statisticsController` y `statsMustache` están desarrolladas de una forma simultánea.

### 3.0 – Recursos

Los recursos empleados en la realización han sido los siguientes.

Recursos humanos	Recursos Fisicos	
Grupo 8	Software	Hardware
	Eclipse	Portatiles
	Microsoft Proyect	Equipos de la empresa
	Spring Tool	

### 4.0 – Mantenimiento

Tras ponernos de acuerdo en los diferentes recursos que necesitamos para implementar hemos decidido añadir las siguientes implementaciones y versiones.

#### Versión 1: v1

La versión 1 del software desarrollado no contiene modificaciones se corresponde con el código del que parte el proyecto que hemos realizado por lo que es el punto de partida del mismo.

- Player.java Versión 2: v2                      Modificación con fecha: 03-Marzo-2020

Se han añadido los siguientes atributos a la clase:

- partidasGanadas
- partidasEmpatadas
- PartidasPerdidas

Así como su inicialización en el constructor. También se han añadido los getters y setters para recuperar dicha información cuando la necesitemos:

- getPartidasGanadas()
- getPartidasPerdidas()
- getPartidasEmpatadas()

Hemos añadido un método *equals* que nos va a servir para comprobar si dos jugadores son iguales al tener el mismo nombre, con esto, podemos llevar un control del jugador que está en la partida y registrar su actividad.

Modificamos el método *toString* para trazar toda la actividad a cada jugador por consola.

- StatisticsService.java Versión: v1                      Modificación con fecha: 05-Marzo-2020

Creación de esta nueva clase para cubrir las necesidades de almacenamiento de los datos referentes al jugador, declaramos un arraylist para almacenar la lista de jugadores. En esta versión hemos creado métodos para añadir jugadores, y actualizar todo lo referentes a las estadísticas que genera el jugador.

- ArrayList<Player> listaJugadores;
- anadirJugador()
- actualizarEmpatesJugador()
- actualizarGanadasJugador()
- actualizarPeridasJugador()

- TicTacToeHandler.java **Versión: v1**      Modificación con fecha: 09-Marzo-2020

En esta primera versión hemos modificado el método handleMessage concretamente en el **case MARK** para comprobar, primeramente si el jugador pertenece a la lista de jugadores posteriormente si se produce un empate lo añadimos a sus estadísticas con el método checkDraw()

- TicTacToeHandler.java **Versión: v2**      Modificación con fecha: 15-Marzo-2020

En esta segunda versión, al tener la certeza que los datos se actualizan correctamente implementamos la posibilidad de que los jugadores ganen y pierdan y sus estadísticas queden reflejadas correctamente. Para ello utilizamos los métodos checkWinner().

Todo ello queda almacenado en el Arraylist de estadísticas que nos hemos declarado que va a ser el que utilicemos después para crear el Template **estadisticas.html**.

- StatisticsService estadisticas = new StatisticsService();

- StatisticsController.java **Versión: v1**      Modificación con fecha: 22-Marzo-2020

Creamos el controlador desde el cual añadimos el modelo para poder llamar al templates estadisticas.html.

- Estadisticas.html      Modificación con fecha: 25-Marzo-2020

Creamos el recurso estadisticas.html con Moustache donde vamos a mostrar las estadísticas que hemos recopilado de los jugadores.

Este recurso es accesible a través de <http://localhost:8080/stats>.