# 1. Autores del trabajo, planificación y entrega

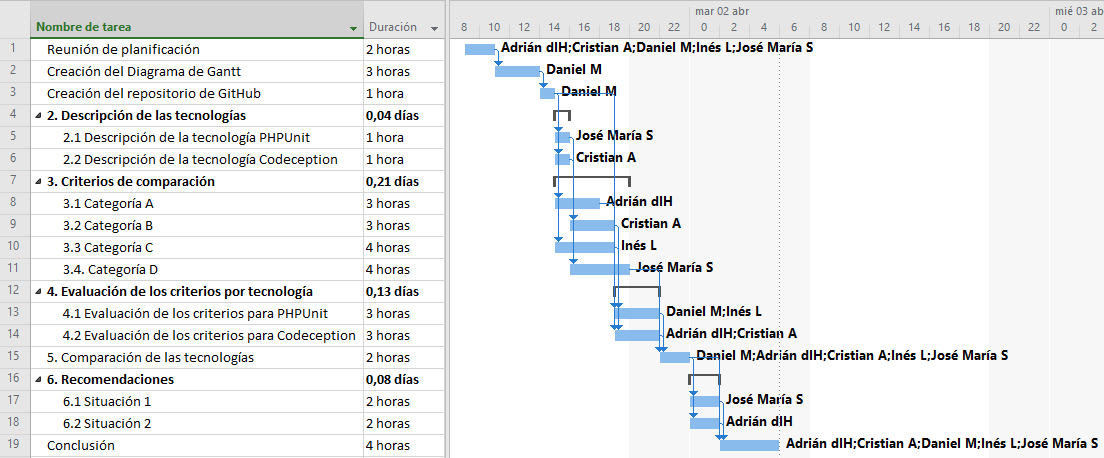
## 1.1 Autores

Grupo T8

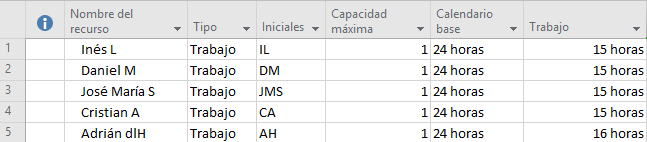
* Daniel Manzano Estébanez (Coordinador)
* Cristián Abellán Madrigal
* Inés López Baldominos
* José María Sanz Górriz
* Adrián de la Hoz Casanova

## 1.2 Planificación

Para establecer la planificación se ha elaborado un diagrama de Gantt con Microsoft Project. Al igual que en el trabajo anterior, cada miembro del grupo debe sumar por lo menos 15 horas de trabajo entre las tareas que tenga asignadas. El diagrama es el siguiente.



La última tarea se corresponde con la unificación y comprobación del trabajo, que es realizada por todos los miembros del grupo.



En esta última imagen se puede apreciar el total de horas que acumula cada participante, quedando los totales balanceados en torno a las 15 horas.

El archivo de Microsoft Project correspondiente a la planificación se encontrará en el repositorio del trabajo, cuya dirección se puede encontrar en el siguiente apartado de este documento.

## 1.3 Entrega

Del mismo modo que en el trabajo TG1, se ha creado un repositorio en GitHub siguiendo las mismas pautas, donde se encontrará la plantilla del trabajo, el trabajo individual de cada miembro y el documento final y presentación del mismo.

El enlace URL es el siguiente: <https://github.com/daniel-manzanoe/TG2>

# 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para la tecnología PHPUnit

|  |  |
| --- | --- |
| CRITERIOS | EVALUACIÓN |
| Criterio A.1: Precio de la herramienta | Gratuita |
| Criterio A.2: Idioma de la herramienta | Inglés |
| Criterio A.3: Popularidad de la herramienta | Alta |
| Criterio A.4: Licencia de la herramienta | BSD |
| Criterio A.5: Documentación disponible | Alta |
| Criterio B.1: Dificultad para la puesta en marcha | Baja |
| Criterio B.2: Prueba de mutación con librería Infection | Si |
| Criterio B.3: Prueba de mutación con librería Humbug | Si |
| Criterio B.4: Análisis de cobertura de código con XDebug | Si |
| Criterio B.5: Velocidad del análisis de cobertura de código | Baja |
| Criterio B.6: Complejidad para analizar los resultados | Media |
| Criterio C.1: Pruebas unitarias | Si |
| Criterio C.2: Pruebas funcionales | Si |
| Criterio C.3: Pruebas de aceptación | Si |
| Criterio C.4: Pruebas de integración | Si |
| Criterio C.5: Pruebas de regresión | Si |
| Criterio C.6: Pruebas de estrés | Si |
| Criterio C.7: Pruebas de rendimiento | Si |
| Criterio D.1: Velocidad pruebas unitarias | Alta |
| Criterio D.2: Velocidad pruebas funcionales | Alta |
| Criterio D.3: Velocidad pruebas de aceptación | Media |
| Criterio D.4: Velocidad pruebas de integración | Baja |
| Criterio D.5: Velocidad pruebas de regresión | Baja |
| Criterio D.6: Velocidad pruebas de estrés | Media |
| Criterio D.7: Velocidad pruebas de rendimiento | Alta |

# 5. Comparación de las tecnologías

|  |  |  |  |
| --- | --- | --- | --- |
| CRITERIOS | PHPUnit | Codeception | COMENTARIOS |
| A.1 | Gratis | Gratis | Se pueden hacer donativos. |
| A.2 | Inglés | Inglés |  |
| A.3 | Alta | Alta |  |
| A.4 | BSD | Open Source | Ambas licencias están relacionadas con el software libre. En el caso de Codeception es MIT license. |
| A.5 | Alta | Media | En el caso de codeception se limita a prácticamente a la documentación oficial. |
| B.1 | Baja | Alta | Codeception requiere una configuración inicial más compleja. |
| B.2 | Si | No | La librería Infection estará próximamente disponible en Codeception. |
| B.3 | Si | Si |  |
| B.4 | Si | Si |  |
| B.5 | Baja | Alta | Un análisis con Codeception puede tardar algo más de 1 min, mientras que en PHPunit estaría en torno a 15 min. |
| B.6 | Media | Baja | Los resultados con codeception son fácilmente interpretables por usuarios sin conocimientos técnicos. |
| C.1 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.2 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.3 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.4 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.5 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.6 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| C.7 | Si | Si | Ambos entornos permiten configurar mediante una plantilla genérica cualquier tipo de pruebas. |
| D.1 | Alta | Alta | Al ser pruebas unitarias (generales) se realizan rápidamente, no se obtienen diferencias notables. |
| D.2 | Alta | Alta | Del mismo modo que en el caso anterior, ambas pruebas funcionales se realizan rápidamente, no encontramos diferencias destacables como para decantarnos por una tecnología u otra por esta característica. |
| D.3 | Baja | Media | En este caso si tenemos una pequeña diferencia de rapidez en las pruebas de aceptación. Son más rápidas en el caso de Codeception. |
| D.4 | Baja | Media | Las pruebas de integración combinan los distintos módulos de una aplicación. Esto es una tarea compleja que depende en cierta medida de como estén codificadas las pruebas. Además, también influye como de optimizadas estén las librerías que use el entorno de pruebas y por lo tanto la velocidad de las mismas está influida por el tipo de entorno que se esté usando. En este caso Codeception usa librerías más optimizadas. |
| D.5 | Baja | Baja | La velocidad de las pruebas de regresión está fuertemente relacionada con lo optimizado que se encuentre el entorno de trabajo que se esté utilizando. Las pruebas de regresión consisten en probar completamente una versión nueva de una aplicación y por lo tanto suele ser un proceso lento. |
| D.6 | Media | Media | Las pruebas de estrés consisten en llevar a condiciones extremas a la aplicación. El gran volumen de datos hace que esta tarea conlleve un tiempo de ejecución medio. Ambos entornos de trabajo están preparados para optimizar el tiempo en este tipo de pruebas. |
| D.7 | Alta | Alta | Las pruebas de rendimiento están relacionadas con las pruebas de estrés. La diferencia es que no suelen llevar a condiciones extremas a la aplicación, sino que se estudia el tiempo de respuesta de la misma. En este sentido, ambos entornos están muy optimizados en este tipo de pruebas. |