

QuarenTEAM

*Nombre del grupo: **QuarenTEAM***

*Nombre del proyecto: **Collateral***

*Nombre del documento: **Informe sobre el trabajo realizado***

Integrantes:

Don Enzo Martin

Quinteros Tomás

Coschica Francisco Nicolás

Hidalgo Juan Nahuel

*Fecha: **23/06/2020***

Índice:

[1 - Esfuerzo dedicado](#)

[2 - Lecciones aprendidas durante la elaboración del proyecto](#)

[3 - Errores cometidos durante la elaboración del proyecto](#)

1- Esfuerzo dedicado:

El esfuerzo realizado para realizar el trabajo en horas/persona por cada integrante del grupo fue:

- Juan Nahuel Hidalgo: 99 horas/persona.
- Don Enzo Martín: 92 horas/persona.
- Tomas Quinteros: 91 horas/persona.
- Coschica, Francisco Nicolás: 90 horas/persona.

Dando un total de 372 horas/persona totales para la implementación del proyecto con toda su documentación asociada.

Consideramos también necesario aclarar que se invirtió un aproximado del 50% más del tiempo antes expuesto aprendiendo a utilizar todas las herramientas dando como resultado un total aproximado de 560 horas/persona.

2- Lecciones aprendidas durante la elaboración del proyecto:

- Es muy difícil hacer una planeación realista para un proyecto cuando se utilizan herramientas que no se conocen.
- Desarrollar juegos o aplicaciones de forma amateur lleva menos tiempo y trabajo. Pero aplicando las practicas de Ingenieria en Software obtenemos un mayor beneficio a largo plazo a pesar del tiempo extra que estas demandan.
- Unity es una gran herramienta para realizar juegos pero eso terminó siendo un percance por el simple hecho de que tuvimos que complejizarlo para responder a las necesidades de la asignatura.
- La herramienta de integración continua Jenkins en conjunto con el motor Unity pese a ser compatibles tienen un grado de dificultad moderado para compaginarse.
- Si hubiésemos aplicado las metodologías ágiles podríamos haber avanzado de manera más efectiva/eficiente.
- Hay que comprobar un correcto funcionamiento de la herramienta de control de versiones, ya que dependiendo del entorno utilizado se pueden encontrar problemas.
- El desconocimiento sobre cómo testear requerimientos no funcionales, puede llevar a que este sea más difícil de lo que se cree. Lo que puede llevar a la planeación de tests que son muy difíciles de implementar.
- Requiere gran planeamiento y disciplina, para elaborar un producto de software siguiendo un esquema de trabajo previo.
- La importancia de realizar los diagramas UML antes de comenzar a escribir el código.
- Los diferentes tipos de test son muy importantes para asegurar que un juego funcione correctamente según se desarrolle.
- La utilización de herramientas para chequear las métricas de un código son importantes para una implementación mantenible y simple del proyecto.

3- Errores cometidos durante la elaboración del proyecto:

- Errores en algunos merges.
- Comits poco descriptivos o redundantes.
- No respetar en algunos casos las reglas de nombramiento para las ramas en el repositorio.
- Requerimientos imposibles de testear.
- No investigar lo suficiente el manejo de los archivos generados por el motor del juego, para su uso con la herramienta de integración continua y la herramienta de control de versiones..
- Usar un motor de juego completamente desconocido que utiliza un lenguaje de programación también desconocido para todos los integrantes, al implementar el proyecto.
- La temprana codificación en el aprendizaje de Unity predispuso al grupo a la hora de realizar el juego, lo cual nos llevó a demoras por reestructuramiento de código.