



# Presentación Final

Ingeniería del Software

**Nombre del grupo:** The Moth

**Nombre del proyecto:** La Mosca

**Nombre del documento:** Informe de la Presentación Final

**Integrantes:**

- Agustín Carranza
- Bruno García
- Emanuel Echazú

**Fecha:** 23/06/2020

**Versión del documento:** 0.2.0

## Versiones del Documento

Versión	Fecha	Resumen del Cambio	Autores
0.1.0	21/06/2020	Creación del documento y desarrollo.	Emanuel Echazú Bruno García Agustín Carranza
0.2.0	23/06/2020	Actualización de Dedicación de esfuerzo.	Emanuel Echazú Bruno García Agustín Carranza

## Índice

[Versiones del Documento](#)

[Índice](#)

[Dedicación de Esfuerzo](#)

[Errores cometidos.](#)

[Lecciones aprendidas](#)

## Dedicación de Esfuerzo

Tarea	Integrantes	Esfuerzo
Implementación de las herramientas de gestión de las configuraciones	Agustín Carranza Bruno García Emanuel Echazú	6 hs 8 hs 9 hs
Diseño y construcción del modelo	Agustín Carranza	28 hs
Diseño y construcción de la vista Partida	Emanuel Echazú	17 hs
Diseño y construcción de la vista Consola	Bruno García	18 hs
Diseño y construcción de controladores	Agustín Carranza Bruno García Emanuel Echazú	15 hs 15 hs 5 hs
Implementación de pruebas	Agustín Carranza	8 hs
Elaboración de documentación referida a manejo de las configuraciones, requerimientos, arquitectura y diseño	Agustín Carranza Bruno García Emanuel Echazú	18 hs 27 hs 30 hs
Elaboración del informe final	Agustín Carranza Bruno García Emanuel Echazú	4 hs 4 hs 4 hs
Elaboración de la presentación	Agustín Carranza Bruno García Emanuel Echazú	5 hs 1 hs 1 hs
	<b>Total</b>	<b>223 hs.</b>

## Errores cometidos.

01. Al momento de la elección del tema del proyecto, el equipo no pudo dimensionar la dificultad y el tiempo que insumiría la implementación de las reglas del juego.
02. De haber elegido una metodología de trabajo con énfasis en el testing, se podrían haber desarrollado un mayor número de pruebas automáticas.
03. Para aprovechar todas las virtudes que significa implementar el patrón de arquitectura MVC, el equipo podría haber diseñado dos modelos (por ejemplo, agregar otro juego de cartas con reglas distintas). Entonces el controlador, usando el patrón de diseño strategy, permitiría elegir en tiempo de ejecución un modelo u otro.

04. En la etapa de gestión de las configuraciones, se optó en un comienzo por la herramienta Jenkins para la integración continua. No se tuvo en cuenta que ésta requiere ser instalada en un servidor dedicado. El equipo no contaba con uno e implementarlo excedía el alcance del proyecto. Finalmente se optó por la herramienta Travis que corre en la nube.

## Lecciones aprendidas

01. Implementar los patrones de diseño strategy y observer no implica haber utilizado el patrón de arquitectura MVC.
02. Se puede implementar el patrón de arquitectura MVC y no utilizar los patrones de diseño antes mencionados.
03. Es muy productivo utilizar patrones que desacoplan funcionalidades ya que permite el desarrollo paralelo, la división de tareas y la optimización de los tiempos.
04. A la hora de estimar tiempos de desarrollo debe tenerse en cuenta las fortalezas y debilidades de los miembros del equipo.
05. Para poder desarrollar un proyecto profesional siguiendo los pasos correctos, en el orden que corresponde y utilizando metodologías acordes, es requisito fundamental tener experiencia en este tipo de trabajo. Si el equipo tuviera que realizar un nuevo proyecto luego de haber culminado las actividades propuestas, seguramente transitaría el desarrollo de todas las etapas de forma más fluida, pudiendo comunicarse con terminología y herramientas ya conocidas, y poniendo foco en el proyecto propiamente dicho.