

# PSP

Irving A. Sanchez R.

## Objetivo:

El siguiente documento tiene el objetivo de informar los métodos organizacionales y técnicas durante el proceso de desarrollo.

## PSP 0

Al momento de desarrollar software se toma en cuenta una lista de funcionalidad a implementar al producto para cumplir su objetivo, estas tareas deben estar organizada en módulos y se selecciona el requisito a implementar en búsqueda de cumplir con la descripción de este en el tiempo establecido (Debido a que se cuenta con poco tiempo, los requisitos solo serán generales, en la búsqueda de optimizar tiempo por lo mismo la fase de tiempo será reducida 4 días de desarrollo). Al final se prueba que el código genere las respuestas esperadas con un protocolo de testeo (En esta ocasión serán pruebas de ejecución de caja negra), en caso de que se encuentre errores se pasan a una lista de correcciones, es por ello por lo que se requiere una holgura antes de la entrega final.

## PSP 0.1

Se describe los usos de ciertas convenciones de programación

Uso paréntesis para que las cláusulas de una expresión sean evidentes, como se muestra en el código siguiente.

-Usar `&&` en vez de `&` y `||` en vez de `|` si no es necesario que verifique ambas posibilidades.

Métodos

Los métodos se escriben en PascalCase.

Fields

- public, public static -> camelCase
- static -> s\_prefix + camelCase
- private, protected, internal -> m\_prefix + camelCase
- const -> UPPER\_CASE

Parámetros

Los parámetros se escriben en camelCase.

Convenciones de los comentarios

- Se coloca el comentario en una línea independiente, no al final de una línea de código.
- Se comienza el texto del comentario con una letra mayúscula.
- Se finaliza el texto del comentario con un punto.
- Se inserta un espacio entre el delimitador de comentario (//) y el texto del comentario.
- Nos se crea bloques con formato de asteriscos alrededor de comentarios.

La explicación de las variables declaradas al inicio (*fields*) estarán comentadas a lado de la variable.

La de los métodos una línea antes de comenzar y comenzando con :

Método [Nombre de metodo]...

Los sets y gets deben estar uno después de otro y solo llevan en la descripción "set y get de [nombre de la variable]"

Las líneas de código arriba de la línea y solo se comenta si realmente necesiten explicación, ejemplo, explicar porque se le aplica una operación a una variable es necesario, explicar que parámetros lleva el constructor de Color no es necesario.

Como propuestas he estipulado que al generar una funcionalidad que solo es para test y no debería estar en la versión final del proyecto en el comentario que lo describa usar la palabra ELIMINAR para facilitar la búsqueda de estas partes del código.

Los comentarios de bloque /\* \*/ solo se utilizan para poner en comentario muchas líneas de código y que no se ejecute.

## PSP 1

Se documenta la planificación estableciendo las tareas necesarias, estas se clasificaran en una tabla de dependencia de actividades, donde se registre los tiempos estimados para cada actividad, así como su prioridad y dependencia, esto sirve para generar una matriz de trazabilidad.

En esta ocasión por tiempo no se generará el diagrama completo, agrupará las actividades en fases. que sustituirán las tareas y se diseña un calendario para finalizarles.

### PSP 1.1

El proceso consistirá en una serie de pasos establecidos tomando en cuenta estándares de calidad, se toman los requerimientos y se seleccionan las adecuaciones para cumplirlos en un periodo de tiempo establecido, cada requerimiento se hará un análisis de modelado de clases con el objetivo de considerar buenas prácticas y convenciones de programación, luego de cubrirse cada sección, se

comenzará con un proceso de documentación, para después, encontrar problemas se agregan los detalles encontrados a una lista de correcciones y optimización.

PSP 2

La gestión de tareas se establece con la metodología Skrum que permite tomar las tareas y gestionar tiempo, para ellos se usara una tabla para las actividades. El protocolo de análisis consistirá en tomar los objetivos y usar modelar con UML los componentes del sistema, se desarrolla código tomando en cuenta los siguientes puntos:

- Priorizar la legibilidad.
- Estructurar la arquitectura.
- Colocar comentarios.
- Simplifica al máximo.
- No reproducir fragmentos idénticos de código.

Se realizarán pruebas unitarias, componentes, integración, sistema, humo y aceptación con una tabla donde se compruebe todo esto.

PSP 2.1

En esta sección se presenta las plantillas de registros de a las actividades.

Trello permitirá gestionar la lista de actividades en tres columnas: lista de tareas, proceso y hecho.

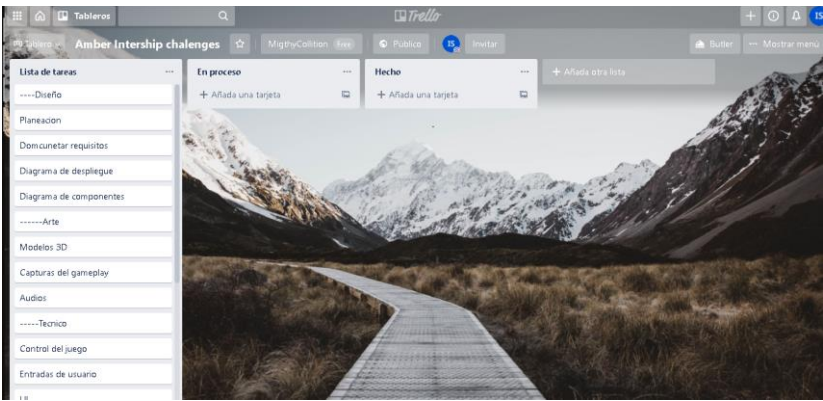


Ilustración 1 Trello para organizar las tareas

Día	dd/mm/aa	dd/mm/aa	dd/mm/aa	dd/mm/aa	dd/mm/aa	dd/mm/aa
	Fase n, fase n+1	Fase n+1...	...	...	..	..

Ilustración 2 calendario

Definición de tareas

Definición de tareas

- 1. Realizar reuniones cada quince días durante una hora.
  - a. Realizar los reportes usando la plantilla para reuniones.
- 2. Generar una lluvia de ideas.
- 3. Generar propuestas.
- 4. Aceptar la propuesta.
- 5. Analizar al público potencial / cliente.
- 6. Analizar al equipo de desarrollo.
  - a. Capacitar al equipo en caso de ser necesario.
- 7. Utilizar la plantilla de productividad por individuo.
- 8. Analizar los requisitos.

Secuencia de actividades

Secuencia de actividades

Actividades	Duración (días)	Dependencia
1	1	-
2	1	M1(1)
3	1	M2(2)
4	1	M3(3)
5	1	M4(4)
6	5	M4(4)
7	1	M1(1)
8	2	M5(5)
9	1	M6(8)