

Proyecto Integrador 2

Documento maestro - Sprint 1

Notas importantes:

- Por cada sprint es necesario tener un repositorio de las ceremonias hechas al interior del equipo, con sus respectivas evidencias (*grabaciones, fotos, actas de reuniones*). Estas evidencias tendrán un peso en la rúbrica de evaluación.
- En este documento se enuncian los contenidos de cada entrega, que serán secciones dentro de una wiki (*o su elemento homólogo en la herramienta de gestión del proyecto que se esté trabajando*). **En esta ocasión NO se solicitará un documento en PDF. Solo el enlace al repositorio y la WIKI.**
- En cada entrega hay una presentación, y cada presentación tiene contenidos asociados y equivalen a un 50% de la nota del Sprint.
- La sustentación de cada entrega es de carácter obligatorio. Ojo: si un integrante no asiste a la sustentación tendrá 0.0 en la nota completa del sprint.

Entrega 2

A continuación, encontrarán todos los ítems evaluables para el **Sprint 1**.

Recuerden que todo esto debe diligenciarse en la plataforma de gestión de proyectos que se esté utilizando. NO se recibe un documento por aparte; **SOLO SE REvisa LA HERRAMIENTA DE GESTIÓN DE PROYECTOS**.

En este Sprint nos centramos en el inicio de la implementación del producto. SE ESPERAN AVANCES A NIVEL FUNCIONAL CON IMPLEMENTACIÓN Y PROTOTIPO FUNCIONAL.

Entregables:

- ✓ Producto software con la funcionalidad comprometida.
- ✓ Actualización de la arquitectura planteada en el sprint anterior (*solo si se considera necesario*).
- ✓ Backlog al día (*¡Es necesario ir cerrando los Sprints!*).
- ✓ Evidencias de las ceremonias.
- ✓ Primera versión del Plan de Negocios.
- ✓ Prototipo Funcional de la Interfaz de Usuario.
- ✓ Casos de pruebas diseñados y ejecutados para el sprint actual.
- ✓ Reporte de errores (*bugs reportados en el backlog y evidencia de la ejecución de casos de prueba*).
- ✓ Checklist calidad del software.
- ✓ Sustentación (*hay sustentación para clientes, y para los profesores*).

Recuerden actualizar los enlaces de los repositorios en caso que hayan cambiado, y dar permisos al profesor del curso.

Entregable	Peso en la nota
Plan de Negocios	15
Casos de Prueba	15
Calidad del software	20

Presentación (Review) mostrando el Desarrollo MVP V1	50
--	----

1. Primera versión del Plan de Negocio

Para el trabajo del Plan de Negocio, deben subir en la Wiki un documento con la estructura que se detalla en el “[Esquema del Plan de Negocio](#)”. Para la entrega del **Sprint 1** sólo deben desarrollar las siguientes secciones de dicho documento:

1. Resumen Ejecutivo
2. Descripción del Negocio
3. Análisis del Mercado
4. Producto o Servicio
5. Implementación

Sección 1. Resumen ejecutivo	
Descripción del Negocio	<ul style="list-style-type: none"> • Claridad y concisión en la descripción del negocio. • ¿Se comprende la propuesta de valor? • ¿Está claro el problema que resuelve el negocio?
Fundadores / Equipo de Gestión	<ul style="list-style-type: none"> • Experiencia relevante del equipo. • Coherencia y complementariedad de habilidades.
Productos / Servicios	<ul style="list-style-type: none"> • Innovación y diferenciación de los productos/servicios. • ¿Están alineados con las necesidades del mercado objetivo?
Mercado Objetivo	<ul style="list-style-type: none"> • Identificación clara del mercado objetivo • Análisis de fortalezas y debilidades comparativas.
Competencia	<ul style="list-style-type: none"> • Claridad y realismo de los objetivos establecidos. • Coherencia entre los objetivos a corto y largo plazo
Objetivos a Corto y Largo Plazo	<ul style="list-style-type: none"> • Claridad y realismo de los objetivos establecidos. • Coherencia entre los objetivos a corto y largo plazo.

Sección 2. Descripción del Negocio	
Historia y Antecedentes	<ul style="list-style-type: none"> • Relevancia de la historia para entender el negocio. • Destacar los hitos importantes en el desarrollo del negocio.
Misión y Visión	<ul style="list-style-type: none"> • Claridad con la propuesta de valor. • Dirección para el negocio a largo plazo.

Valores	<ul style="list-style-type: none"> • Coherencia entre los valores declarados y la operativa del negocio. • ¿Se reflejan los valores en la cultura empresarial?
----------------	--

Sección 3. Análisis del Mercado	
Investigación de mercado	<ul style="list-style-type: none"> • Profundidad y amplitud de la investigación. • ¿Se muestran fuentes fiables y actualizadas?
Segmentación del Mercado	<ul style="list-style-type: none"> • Claridad en la segmentación del mercado. • ¿Se identifican segmentos específicos y sus necesidades?
Clientes Objetivo	<ul style="list-style-type: none"> • Comprensión profunda de los clientes objetivo. • ¿Se muestran <i>insights</i> relevantes sobre sus comportamientos y preferencias?
Análisis de Competencia	<ul style="list-style-type: none"> • Complejidad y detalle del análisis de competencia. • ¿Se identifican puntos fuertes y débiles de los competidores?

Sección 4. Producto o servicio	
Descripción del Producto / Servicio	<ul style="list-style-type: none"> • Claridad en la descripción de características y funcionalidades. • ¿Se destacan los beneficios clave para el cliente?
Propuesta de Valor	<ul style="list-style-type: none"> • Claridad en la propuesta de valor única. • ¿Se muestra cómo el producto/servicio resuelve el problema del cliente?
Ciclo de Vida del Producto	<ul style="list-style-type: none"> • Comprensión del ciclo de vida del producto / servicio. • ¿Se proponen estrategias para cada etapa del ciclo de vida?

Sección 5. Implementación	
Cronograma	<ul style="list-style-type: none"> • Realismo y detalle del cronograma. • ¿Se identifican hitos clave y se asignan plazos realistas?
Hitos Clave	<ul style="list-style-type: none"> • Importancia y coherencia de los hitos establecidos. • ¿Son medibles y alcanzables?

Recursos Necesarios	<ul style="list-style-type: none"> • Identificación clara de recursos necesarios. • ¿Se justifica adecuadamente cada recurso?
----------------------------	---

2. Desarrollo de código

Entregable asociado: Un desarrollo funcional mínimo viable (MVP) que contenga las funcionalidades de la solución planeadas en el Sprint 1 en el repositorio.

Desarrollo Sprint1	
Cumplimiento funcional	<ul style="list-style-type: none"> • ¿Las historias de usuario planeadas para el Sprint 1 fueron implementadas correctamente en el MVP1?
Gestión técnica del desarrollo	<ul style="list-style-type: none"> • ¿Se usaron adecuadamente las herramientas de desarrollo como git? • ¿El repositorio contiene todo el código correspondiente al MVP 1?

3. Diseño de casos de Pruebas Funcionales

Casos de pruebas (CP)	
Casos de pruebas	<ul style="list-style-type: none"> • Las historias de usuario (HU) del Sprint 1 tienen criterios de aceptación y CP funcional asociados en la herramienta de gestión de proyectos. • Los CP se insertan asociados a las historias de usuario y a sus criterios de aceptación. • Los criterios de aceptación se redactan de forma similar al siguiente ejemplo: <i>“Dada la situación xxxx, cuándo se produce el xxxx acontecimiento debe pasar xxxxx”</i>.
Ejecución de pruebas y errores	<ul style="list-style-type: none"> • ¿Hay alguna evidencia de que los casos de prueba diseñados se hayan ejecutado? • ¿hay reporte de errores? Los errores deben verse reflejados como bugs en la herramienta de gestión de proyectos en caso de encontrar alguno al ejecutar los casos de prueba, cuya ejecución debe estar igualmente documentada.
Conexión HU-CP-Errores	<ul style="list-style-type: none"> • ¿Se evidencia conexión lógica entre historia de usuario, caso de prueba y reporte de error?

	*Los casos de prueba deben de estar asociados a los requisitos, y no probar cualquier cosa solo por probar.
--	---

4. Checklist Calidad del Software

o Estándares de nombramiento

- El proyecto debe de seguir un estándar de nombramiento claro y conciso.
- Este debe de estar documentado en la wiki, al igual que una justificación del mismo. Idealmente, se deberá seguir un estándar oficial del lenguaje o *framework*, o un estándar reconocido por la comunidad. Agregar las debidas referencias.

● Análisis estático de código

- El proyecto debe de usar al menos una herramienta de análisis estático de código.
- Se deberá configurar la herramienta al gusto del equipo de trabajo; se recomienda iniciar con un modo estricto o un modo recomendado por la comunidad.
- La(s) herramienta(s) a utilizar deberá ser documentada(s) en la wiki, al igual que una justificación de su elección y configuración.
- Puntos adicionales si los estándares de nombramiento son enforzados por la(s) herramienta(s) de análisis estático de código.
- Puntos adicionales si la herramienta soporta corrección automática de errores estéticos, también conocidos como formateadores de código.

● Extras

- Seguir una estrategia de *branching* para Git, como: GitFlow, Github Flow, *Trunk Based Development*, etc. Documentar en la wiki la estrategia usada y su justificación.
- Tener un *pipeline* de integración continua que automatice el proceso de calidad. Conectar dicho *pipeline* con el repositorio de código, y bloquear la fusión de un *pull request* hasta que el *pipeline* este “*green*” y tenga un *approve* (aprobado) de revisión de pares.
- Herramientas de análisis de seguridad y vulnerabilidades.

Calidad del software	
Estándar de nombramiento	<ul style="list-style-type: none"> • El proyecto sigue un estándar oficial del lenguaje/framework/comunidad? • ¿Se documentó en la Wiki el estándar seguido y su justificación?
Análisis estático de código	<ul style="list-style-type: none"> • ¿Se utiliza al menos una herramienta de análisis estático correctamente integrada al flujo de trabajo y configurada por el equipo? • ¿Se documentó en la Wiki la configuración usada para la(s) herramienta(s) y su justificación?

5. Sustentación

Es de carácter obligatorio para todos los integrantes.

Si un integrante no asiste a la sustentación tendrá una nota de 0,0 EN EL SPRINT.

Una *review* convencional donde se muestra el *sprint goal*, los avances y las deudas o elementos que quedaron pendientes. Se sugiere que sea tipo *pitch*.