

PENSAMIENTO COMPUTACIONAL

Proyecto Semestral de Software

Guía del proyecto + pauta de Presentación 1 + pauta de Presentación 2 (Final)

El desafío del semestre

Pasar de una idea a un producto de software demostrable: identificar un problema, diseñar una solución, modelarla, planificarla, programar un MVP, trabajar con datos y defender profesionalmente las decisiones del equipo.

Este documento reúne en un solo lugar el recorrido del proyecto y los criterios de las dos presentaciones. La Presentación 1 valida el diseño; la Presentación 2 valida la construcción y el funcionamiento del MVP.

¿Qué hemos estado construyendo?

Durante el semestre cada grupo ha trabajado como un pequeño equipo de desarrollo de software.

El proyecto no consiste solamente en programar. El objetivo es recorrer un proceso completo: entender una necesidad, convertirla en requerimientos, modelar el sistema, organizar el trabajo, construir una solución y demostrar que esa solución funciona.

Recorrido del semestre

- 1. Problema y oportunidad** Definir qué situación quieren resolver o mejorar, quién recibe el valor y por qué el proyecto importa.
- 2. Diseño de la solución** Definir objetivo, propuesta de valor, funcionalidades, alcance del MVP e indicadores de éxito.
- 3. Organización del trabajo** Usar backlog, prioridades, roles, tareas y una lógica de trabajo iterativa inspirada en metodologías ágiles.
- 4. Modelamiento** Representar el sistema mediante diagramas de flujo/casos de uso, secuencia y base de datos o clases.
- 5. Programación** Construir interfaces y lógica del producto, probar funciones y corregir errores.
- 6. Datos** Diseñar cómo se guarda la información, trabajar con tablas, relaciones y una base de datos cuando corresponda.
- 7. MVP** Integrar los componentes indispensables para que el usuario pueda completar el flujo principal del producto.
- 8. Comunicación profesional** Presentar, demostrar y defender el proyecto frente a preguntas, justificando decisiones técnicas y de producto.

Idea central

La nota final no depende de que el producto sea enorme. Un MVP pequeño, coherente, funcional y bien explicado vale más que una idea gigante que no puede demostrarse.

ETAPAS

De la idea al MVP

Las dos presentaciones representan dos momentos distintos del mismo proyecto.

PRESENTACIÓN 1

Diseñar y demostrar que el proyecto tiene sentido.

- Problema, usuario y objetivo
- Valor e innovación
- Backlog y evidencia
- KPI
- Diagramas técnicos
- Defensa del diseño

PRESENTACIÓN 2

Construir y demostrar que el producto funciona.

- MVP funcional
- Persistencia de datos
- Integración técnica
- Pruebas y estabilidad
- Demostración en vivo
- Defensa del producto

Qué significa “MVP”

MVP = Producto Mínimo Viable. Es la versión más pequeña del producto que permite demostrar su propuesta de valor con un flujo principal real. No necesita tener todas las funciones imaginadas, pero lo que el equipo declara como núcleo debe poder probarse.

- El usuario puede iniciar y completar al menos el proceso principal del producto.
- Las funciones esenciales están conectadas; no son solamente pantallas aisladas.
- Los datos relevantes se guardan y se recuperan cuando el proyecto lo requiere.
- El grupo conoce sus limitaciones y puede explicar qué quedaría para una siguiente versión.

PRESENTACIÓN 1

Diseño del Proyecto de Software

Objetivo: convencer a un posible inversor de que la solución tiene propósito, valor y un diseño coherente.

Qué debe contener

- Problema u oportunidad: qué situación quieren mejorar, resolver o transformar.
- Objetivo del proyecto: qué debería lograr el software.
- Usuario o beneficiario: quién usa el sistema y quién recibe el valor.
- Generación de valor: qué cambia gracias a la solución.
- Innovación o diferenciación: qué hace interesante la propuesta.
- KPI: indicadores que permitirán saber si el proyecto cumple su objetivo.
- Backlog y evidencia: tareas, prioridades, decisiones, cambios y evolución del trabajo.
- Diagramas coherentes: casos de uso/flujo, secuencia y base de datos/clases.

Pregunta que debe poder responder el equipo

Si mañana otra persona tuviera que comenzar a programar su proyecto, ¿la documentación le permitiría entender qué debe construir?

Durante la defensa

Todos los integrantes deben participar y demostrar dominio del proyecto. No basta con mostrar diapositivas: el equipo debe explicar por qué tomó sus decisiones y responder preguntas con coherencia.

EVALUACIÓN

Pauta - Presentación 1

Diseño del Proyecto de Software · Puntaje total: 100 puntos

Criterio	Qué se observará	Pts
Problema, usuario y objetivo	La propuesta se entiende con claridad y existe relación entre necesidad, usuario y objetivo.	15
Valor e innovación	El equipo explica qué valor genera y qué hace interesante o diferente su idea.	15
Backlog y evidencia	Existe planificación y se demuestra trabajo real, decisiones y evolución del diseño.	15
KPI	Los indicadores permitirían comprobar si el proyecto cumple su objetivo.	10
Casos de uso / flujo	Representa correctamente actores, acciones y recorrido principal del sistema.	15
Secuencia	Explica correctamente la interacción y el orden de un proceso relevante.	10
Base de datos / clases	La estructura propuesta es coherente con las funciones y datos del sistema.	15
Presentación y defensa	El grupo comunica con claridad y puede justificar sus decisiones.	5

Antes de presentar

- ¿Un tercero entiende la idea?
- ¿Se ve claramente el valor?
- ¿Los diagramas hablan del mismo sistema?
- ¿Tenemos evidencia del proceso?
- ¿Podemos defender por qué tomamos cada decisión?

PRESENTACIÓN 2

Presentación Final - MVP

Objetivo: demostrar que el proyecto pasó del diseño a un producto funcional y que el equipo entiende cómo está construido.

Lo que debe demostrar el equipo

- El MVP funciona y permite ejecutar el flujo principal del producto.
- Existe coherencia entre lo prometido en la Presentación 1 y lo construido; los cambios están justificados.
- El sistema trabaja con datos de manera ordenada. Cuando corresponda, utiliza una base de datos real y relaciones coherentes.
- Las partes del proyecto se integran: interfaz, lógica, backend/base de datos u otros componentes según el caso.
- El equipo probó su solución, detectó errores y puede explicar qué corrigió.
- Todos los integrantes comprenden el producto y pueden responder preguntas técnicas y de diseño.

No se exige un producto comercial terminado

Se evalúa un MVP. Pueden existir funciones pendientes, mejoras visuales o ideas futuras. Lo importante es que el núcleo del producto sea demostrable y que el equipo pueda explicar con precisión qué funciona, qué no y por qué.

Uso de herramientas e IA

Pueden utilizar herramientas de apoyo durante el desarrollo. En la evaluación, sin embargo, cada grupo debe entender su solución. Si una parte fue generada o asistida por IA, deben ser capaces de explicar qué hace, cómo se conecta con el resto del sistema y qué decisiones tomaron al integrarla.

GUION RECOMENDADO

¿Cómo presentar el MVP?

La demostración debe ser breve, ordenada y centrada en evidencia real.

- 1. Recordatorio del problema** - En 30-60 segundos recuerden qué problema resuelve el proyecto, para quién y cuál es su propuesta de valor.
- 2. Qué se construyó** - Expliquen qué funciones forman parte del MVP y qué cambió respecto del diseño inicial.
- 3. Demo en vivo** - Usen el producto como lo usaría una persona real. Muestren el flujo principal completo, no una colección de capturas.
- 4. Datos y arquitectura** - Expliquen brevemente dónde se guardan los datos y cómo se conectan las partes principales del sistema.
- 5. Pruebas y problemas** - Muestren al menos un ejemplo de error, dificultad o cambio que hayan tenido que resolver durante el desarrollo.
- 6. Resultado y próximos pasos** - Relacionen el MVP con el objetivo/KPI inicial y expliquen qué construirían después.
- 7. Defensa** - Respondan preguntas. Todos los integrantes deben estar preparados para participar.

Evidencia recomendada

- Aplicación funcionando en el computador.
- Base de datos con registros reales de prueba.
- Backlog o tablero actualizado.
- Capturas de versiones anteriores o errores solucionados.
- Código o estructura del proyecto solo cuando ayude a explicar una decisión.

EVALUACIÓN

Pauta - Presentación 2 (Final)

Demostración del MVP · Puntaje total: 100 puntos

Criterio	Qué se observará	Pts
MVP funcional	El flujo principal puede ejecutarse de principio a fin y las funciones esenciales entregan un resultado real.	25
Datos y persistencia	La información se guarda, consulta y relaciona de forma coherente. La estructura de datos responde a las necesidades del producto.	15
Integración técnica	Las partes principales del sistema se conectan correctamente: interfaz, lógica, backend, base de datos o servicios utilizados.	10
Cumplimiento del diseño	Existe coherencia con los requerimientos y diagramas de la Presentación 1, o el equipo justifica claramente los cambios realizados.	10
Experiencia de uso	El MVP puede ser utilizado con claridad; la interfaz y los mensajes permiten entender qué hacer y qué ocurrió.	10
Pruebas y estabilidad	El equipo demuestra pruebas, manejo de errores y correcciones. El producto se comporta de forma suficientemente estable durante la demo.	10
Proceso y evidencia	Se observa evolución real del proyecto mediante backlog, versiones, decisiones, problemas resueltos o mejoras realizadas.	5
Resultado y próximos pasos	El equipo conecta el MVP con el objetivo/KPI inicial y distingue con claridad lo logrado de lo que quedaría	5

CHECKLIST FINAL

Antes de la Presentación 2

Utilicen esta lista durante las últimas clases para decidir qué falta y qué deben priorizar.

- ☐ Podemos abrir el producto y comenzar la demo sin depender de configuraciones improvisadas.
- ☐ El flujo principal funciona de principio a fin.
- ☐ Las funciones esenciales del MVP están conectadas.
- ☐ Tenemos datos de prueba suficientes para demostrar el sistema.
- ☐ Sabemos explicar las tablas, relaciones o estructura de datos utilizada.
- ☐ Podemos explicar cómo se conectan las partes principales del proyecto.
- ☐ Revisamos errores evidentes y probamos el producto antes de presentar.
- ☐ Actualizamos el backlog y podemos mostrar cómo evolucionó el proyecto.
- ☐ Podemos explicar al menos un problema técnico que resolvimos.
- ☐ Sabemos qué cambió respecto de la Presentación 1 y por qué.
- ☐ Todos los integrantes saben usar y explicar el producto.
- ☐ Tenemos claro qué haríamos en una siguiente versión.

Regla de prioridad

Si queda poco tiempo, prioricen que el flujo principal funcione. Después mejoren diseño visual y funciones secundarias. Un MVP estable y demostrable debe ser la meta.

Cierre del proyecto

La Presentación 1 preguntaba: “¿Este proyecto merece ser construido y sabemos cómo hacerlo?”. La Presentación 2 pregunta: “¿Fuimos capaces de convertir ese diseño en una solución que funciona?”. Esa transición -de pensar a construir, probar y explicar- es el objetivo central del proyecto semestral.