Entrega 2 (E2) — Implementación Frontend

Proyecto semestral · Segunda de tres entregas

Fecha de entrega: 26 de mayo, hasta las 23:59

1. Contexto

La Entrega 2 es la continuación natural de la E1 (Design Thinking). Mientras en E1 exploraron el problema, entendieron al usuario y definieron una propuesta de solución, en E2 deben materializar esa propuesta como un producto digital funcional implementado en el frontend.

Esta entrega evalúa dos dimensiones en conjunto: la calidad del diseño (UI design e interaction design) y las buenas prácticas de desarrollo en React / Next.js vistas en clase. No se evalúa por separado, porque en esta disciplina las decisiones de diseño y de implementación están acopladas: un buen patrón de Gestalt mal renderizado pierde su propósito, y una arquitectura de componentes impecable sobre una interfaz confusa tampoco sirve.

2. Objetivo de la entrega

Construir una aplicación web que resuelva de manera concreta y demostrable los pains y gains principales identificados en la E1, combinando decisiones de diseño justificadas con una implementación técnica sólida en React / Next.js.

El foco de esta entrega es el frontend. No se espera backend propio: todos los datos externos pueden ser mockeados (archivos JSON locales, datos estáticos, respuestas simuladas, handlers de Next.js que devuelvan datos fijos, etc.). Lo que se evalúa es qué tan bien está resuelta la experiencia del usuario a nivel de interfaz y qué tan bien aplicado está lo visto en clase.

3. Alcance y requerimientos

3.1 Requerimientos de producto

- La aplicación debe cubrir los flujos principales que resuelven los pains y gains declarados. No se exige cubrir absolutamente toda la idea original: se exige cubrir bien lo que se propone.
- Todos los datos externos pueden estar mockeados. Si se simula una API, el mock debe ser consistente: misma forma de datos, estados de carga, estados de error, estados vacíos.
- La aplicación debe ser navegable de punta a punta sin callejones sin salida. Cada pantalla alcanzable debe estar implementada (no wireframes estáticos ni "coming soon").
- Debe funcionar correctamente en desktop. El soporte mobile es deseable pero no obligatorio salvo que la naturaleza del producto lo exija (ej.: si en E1 definieron una experiencia mobile-first).

3.2 Requerimientos de diseño (UI + Interaction)

Se espera que las decisiones de diseño sean intencionales y justificables. No se pide solamente "algo bonito"; se pide algo coherente y fundamentado en lo visto en clase.

- Aplicación consistente de principios visuales: uso del espacio, contraste, jerarquía, punto focal.
- Uso explícito y reconocible de patrones de Gestalt (proximidad, similitud, continuidad, cierre, figura/fondo) donde aporten a la legibilidad de la interfaz.
- Estados de interacción claros: hover, focus, active, disabled, loading, empty, error. No se evalúa que estén todos en todos lados, pero sí que estén donde corresponde.
- Aplicación del progressive disclosure con contenido visto en clase.

3.3 Requerimientos técnicos (React / Next.js)

Se evalúa el uso correcto de los conceptos y patrones vistos en clase. La decisión técnica importa más que el volumen de código.

- Proyecto implementado en Next.js (App Router) con TypeScript.
- Estructura de carpetas clara y consistente. Componentes organizados por responsabilidad.
- Uso apropiado de Server Components y Client Components.
- Uso correcto de hooks (useState, useEffect, useContext, hooks customizados cuando corresponda). No se evalúa la cantidad, se evalúa la pertinencia.
- Estrategia de rendering consistente con el caso de uso de cada vista: estática, SSR, client-side, o mezcla. Los grupos deben ser capaces de explicar por qué eligieron cada estrategia.
- Gestión de estado razonable para el tamaño del proyecto.
- Componentes reutilizables cuando tiene sentido; componentes específicos cuando tiene sentido.
- Código legible: nombres descriptivos, funciones con una responsabilidad, ausencia de código muerto o comentado "por si acaso".
- Para los estilos deben usar Tailwind CSS.
- Si el equipo decide usar React puro (en lugar de Next.js), deberá justificar en el informe adjunto las razones detrás de la decisión e identificar al menos 2 trade-offs asociados a esa elección (ej.: pérdida de SSR, peor SEO, ausencia de file-based routing, manejo manual de code-splitting, etc.). La justificación debe conectarse con las necesidades reales del producto, no ser una preferencia personal.

4. Informe acompañante

Junto con el código, se debe entregar un informe en formato PDF (máximo 8 páginas, sin contar portada ni anexos) que contenga:

4.1 Pains y gains resueltos

Sección central del informe. Deben listar al menos 4 puntos en total (combinando pains y gains) y, para cada uno, explicar concretamente cómo la entrega lo resuelve.

Para cada pain o gain se espera:

- Descripción breve del pain o gain, y porque este es fundamental (heredado de E1, se permite refinar).
- Decisión de producto tomada para resolverlo (qué feature, qué flujo, qué pantalla).
- Decisión de diseño asociada (qué principio visual o patrón de interacción se aplicó y por qué).
- Referencia concreta a la implementación: nombre de la vista, ruta, o componente donde se materializa. Idealmente con una captura de pantalla.

4.2 Style Guide del proyecto

Cada equipo debe definir y entregar un style guide propio del proyecto. Esto NO es un design system: no se pide documentar variantes, tokens versionados, ni una librería de componentes reutilizables. Es un acuerdo visual mínimo del grupo, que sirve como referencia para mantener consistencia durante el desarrollo y como base contra la cual se evaluará la UI implementada.

El style guide debe incluir, como mínimo:

- **Paleta de colores.** Colores principales, secundarios, y de estado (éxito, error, advertencia, neutro). Indicar para qué se usa cada uno.
- **Tipografía.** Familia(s) tipográfica(s) y escala usada (tamaños y pesos para titulares, subtítulos, cuerpo, captions) .
- **Espaciado y grilla.** Escala de espaciado (ej.: múltiplos de 4 u 8) y, si aplica, sistema de grilla o anchos máximos de contenido.
- **Estilo de bordes y elevación.** Radio de bordes (border-radius) y uso de sombras o bordes para separar superficies.
- **Estilos de elementos clave.** Cómo se ven los botones (primario, secundario, deshabilitado), inputs (default, focus, error), y links. Adjunta ejemplos visuales
- **Iconos.** Librería o set de iconos elegida (ej.: lucide-react, heroicons, material icons), estilo (line vs. filled), y tamaños usados según contexto. Deben mantener un único estilo a lo largo de la aplicación.
- **Tono y voz.** Lineamiento breve sobre cómo le habla el producto al usuario (ej.: cercano vs. formal, uso de tú/usted, mayúsculas en titulares).

El style guide puede entregarse como una sección del informe (con muestras visuales) o como un anexo. Lo importante es que esté declarado de forma explícita: la UI implementada se evaluará contra lo que el equipo se comprometió a seguir.

Tip: Pueden utilizar las herramientas las vistas en clase para asistir en ese proceso.

4.3 Decisiones de diseño

Justificacíon de minimo 3 decisiones de diseño más importante que están detrás de las pantallas: jerarquía, patrones de Gestalt aplicados, principios de interacción (incluyendo progressive disclosure cuando aplique). No se pide un manual de marca: se pide que las decisiones sean conscientes y se puedan defender.

4.4 Decisiones técnicas

Resumen de las decisiones técnicas relevantes: estructura del proyecto, estrategias de rendering utilizadas por vista, cómo se manejan los datos mockeados, uso de librerías externas.

4.5 Estado de avance y próximos pasos

Qué quedó implementado, qué quedó fuera de alcance, y qué planean para la E3.

5. Entregables

- **Código en Zip, recordar no subir node_modules**.
- **Informe en PDF** con el contenido descrito en la sección 4.

Ambos entregables se suben a Canvas a más tardar el 24 de mayo a las 23:59. El commit considerado para evaluación será el último realizado antes del deadline.

6. Entrevista de evaluación (2 de junio)

En la fecha correspondiente a la I2 se realizará una entrevista por grupo. Esta entrevista reemplaza las presentaciones tradicionales y es parte obligatoria de la evaluación.

6.1 Condiciones

- Todos los integrantes del grupo deben estar presentes. La ausencia de un integrante sin justificación formal afectará la nota de ese integrante.
- Duración aproximada: 20–30 minutos por grupo.
- Las preguntas pueden ser dirigidas (a un integrante específico) o abiertas (al grupo). Se espera que todos los integrantes puedan responder sobre cualquier parte del proyecto.
- Todos los grupos deben estar presentes en la sala en la fecha y horaria del I2.

6.2 Qué se evalúa en la entrevista

La entrevista controla dos dimensiones:

a) Entendimiento del proyecto (idea + código).

Se espera que el grupo pueda explicar qué hace su producto, a quién resuelve el problema, y dónde en el código vive cada pieza clave. Ejemplos de preguntas:

- Cuéntame cómo tu producto resuelve el pain principal del usuario.
- ¿En qué parte del código está implementado este feature?

b) Dominio de la materia vista en clase.

Se espera que el grupo pueda conectar las decisiones tomadas con los conceptos vistos en clase. Ejemplos de preguntas:

- ¿Qué estrategia de rendering aplicaste en esta vista y por qué?
- ¿Qué patrón de Gestalt aplicaron acá?
- ¿Por qué este componente es Client Component y no Server Component?

6.3 Efecto de la entrevista en la nota

La entrevista puede modificar la nota obtenida por el informe y el código:

- Incremento: si el grupo demuestra un manejo óptimo de su proyecto y de la materia, con respuestas sólidas, justificadas y consistentes entre integrantes.
- Neutro: si el grupo responde correctamente a las preguntas básicas sobre su proyecto y la materia, sin destacarse ni mostrar carencias evidentes. La nota base de la entrega no se modifica.
- Decremento: si se observa falta de manejo, desconocimiento del propio código, o incapacidad de conectar decisiones con los conceptos del curso.

El efecto de la entrevista sobre la nota se aplica de la misma forma a todos los integrantes del grupo, sin distinción por desempeño individual durante la instancia. La única excepción es la ausencia a la entrevista: el integrante ausente no recibe el ajuste del grupo y su nota se evalúa por separado. Si la ausencia no está debidamente justificada de acuerdo a las condiciones académicas vigentes de la universidad, la nota máxima del integrante ausente en esta entrega será 4,0.

7. Criterios de evaluación

La nota de la E2 se construye a partir de los siguientes criterios, cuyo puntaje suma 100 puntos. La conversión a nota chilena se detalla en la sección 8.

Resolución de pains y gains (20 pts)

- Los pains y gains declarados están efectivamente resueltos por la aplicación.
- La conexión entre problema, decisión de diseño, y resultado es clara y demostrable.

Calidad de diseño (UI + Interaction) (30 pts)

- Coherencia visual en toda la aplicación, evaluada contra el style guide propuesto por el equipo. Se penaliza cuando la UI implementada se desvía del style guide declarado sin justificación.
- Aplicación fundamentada de principios de diseño y patrones de Gestalt.
- Estados de interacción bien resueltos. Feedback apropiado al usuario.

Calidad técnica (React / Next.js) (30 pts)

- Uso apropiado de Server / Client Components.
- Estrategias de rendering justificadas y consistentes.
- Uso correcto de hooks y gestión de estado.
- Estructura de proyecto, legibilidad de código, y calidad del historial de commits.

Informe (20 pts)

- Cumple con todas las secciones requeridas.
- Las justificaciones son concretas y verificables, no genéricas.

Entrevista

- Ajuste positivo o negativo según lo descrito en la sección 6.3.

8. Cálculo de la nota

La nota final de la entrega se calcula con la siguiente fórmula:

Nota final = (puntaje obtenido / 100) × nota máxima + ajuste por entrevista

Donde:

- **Puntaje obtenido:** suma de los puntos obtenidos en los criterios de la sección 7 (sobre un total de 100 pts).
- **Nota máxima:** 7,0 en condiciones normales. Si el integrante no asiste a la entrevista sin justificación válida, su nota máxima en esta entrega es 4,0 (ver sección 6.3).
- **Ajuste por entrevista:** incremento o decremento aplicado según el desempeño del grupo en la entrevista (ver sección 6.3).

Ejemplo: un grupo que obtiene 85 puntos en los criterios y un ajuste positivo de +0,3 en la entrevista, tendría una nota final de (85/100) × 7,0 + 0,3 = 5,95 + 0,3 = 6,25.

9. Política de atrasos y grupos

- Grupos de 2 a 3 integrantes.
- Se aplican los 4 días libres de atraso del semestre. Una vez consumidos, cada día de atraso equivale a un descuento de 0,5 puntos sobre la nota final de la entrega.
- La entrevista no admite atraso: se realiza en la fecha agendada por el equipo docente.

10. Recomendaciones

- Prioricen profundidad sobre amplitud. Es mejor 3 flujos bien resueltos que 10 flujos a medias.
- Todos los integrantes deben tocar código y diseño. La especialización extrema penaliza en la entrevista.
- Anoten las decisiones técnicas mientras las toman, no al final. La memoria no aguanta hasta la entrevista.