



Guía para el profesor

Producción de Diseño Interactivo

LSAN18o8



Índice

Información general del curso	1
Metodología	2
Evaluación	3
Bibliografía	4
Tips importantes	5
Notas de enseñanza	6
Evidencia	13

Información general del curso

Modalidades

- Clave banner: LSAN18o8
- Modalidad: Presencial

Competencia del curso

Diseña, desarrolla e implementa soluciones interactivas que mejoren la experiencia del usuario, utilizando herramientas y principios de diseño de interacción, considerando la accesibilidad, la usabilidad, la interacción multimodal y la integración con tecnologías web actuales.





Metodología

Este proyecto consiste en planificar el diseño de la aplicación, definiendo su propósito, público objetivo y los requisitos técnicos que se implementarán. Además, los estudiantes desarrollarán wireframes y prototipos de alta fidelidad, estableciendo la estructura de navegación y la jerarquía de la información. El proyecto se divide en cuatro fases, en las que deberás reflexionar y trabajar arduamente. Las fases del proyecto son las siguientes:

- **Avance 1.** Elaborarás un documento donde describas el propósito de la aplicación y los objetivos de la interfaz, detallando la audiencia objetivo y los requisitos técnicos.
- **Avance 2.** Implementarás la funcionalidad básica de la aplicación, integrando la geolocalización en tiempo real y la visualización interactiva de mapas, además de personalizar el contenido basado en la ubicación del usuario.
- **Avance 3.** Optimizarás el rendimiento de la aplicación, mejorarás su accesibilidad y realizarás pruebas de usabilidad para garantizar una experiencia de usuario fluida y accesible.
- **Entrega final.** Presentarás la aplicación web completa, mostrando todas las funcionalidades implementadas, como la geolocalización, la visualización de mapas, la personalización del contenido y la adaptabilidad en diferentes dispositivos. Debe demostrar cómo la aplicación cumple con los requisitos de diseño, accesibilidad y rendimiento.

Nota importante: deberás realizar cada una de las fases de manera consecutiva. Para conseguir la acreditación de este certificado, deberás entregar cada una de las cuatro fases, según las indicaciones que se te proporcionan. Omitir alguna de las cuatro fases llevará a la no acreditación del certificado.

Fases

El proyecto está segmentado en **cuatro fases** que abarcan las **cuatro semanas** de trabajo, según esta distribución:

Fase	Actividad de aprendizaje	Producto
1	Planificación y diseño de la aplicación web interactiva	Documento de Word o PDF
2	Desarrollo del prototipo interactivo con geolocalización y personalización de contenidos	Código en Google colab
3	Optimización y pruebas de usabilidad y accesibilidad de la aplicación web	Word o PDF que contenga los resultados de pruebas de usabilidad y accesibilidad Código en Google Colab
4	Entrega final del proyecto de aplicación web interactiva con geolocalización y diseño adaptable	Entrega el código completo de la aplicación junto con la documentación técnica que explique cómo funcionan la aplicación, los componentes clave y las decisiones de diseño.



Evaluación

Unidades	Instrumento evaluador	Porcentaje
1	Avance I	10
1	Avance II	20
1	Avance III	40
1	Entrega final	30
Total		100 puntos



Bibliografía

Libros de apoyo

- ➔ Rogers, Y., Helen, H., y Preece, J. (2023). *Interaction Design: Beyond Human-Computer Interaction* (6ª ed.). EE. UU.: Wiley.
ISBN: 978-1119901099
- ➔ Farahi, B., y Leach, N. (2023). *Interactive Design: Towards a Responsive Environment*. Suiza: Birkhäuser.
ISBN: 978-3035626889



 **Tips importantes**

El alumno deberá realizar cada una de las fases de manera consecutiva. Para conseguir la acreditación de este certificado, deberá entregar cada una de las cuatro fases, según las indicaciones que se proporcionan. Omitir alguna de las cuatro fases llevará a la no acreditación del certificado.



Notas de enseñanza

Fase I Planificación y diseño de la aplicación web interactiva

1. Investigación y planificación

Propósito y objetivos

- **Propósito de la aplicación:** explica qué problema resuelve o qué necesidad satisface. Por ejemplo, "Una app que muestra cafeterías cercanas con wifi gratuito".
- **Objetivos de la interfaz:** claridad, facilidad de uso, rapidez de acceso a la información clave, y adaptación al contexto de uso (móvil o escritorio).

Consejo didáctico: enfatiza la importancia de entender el "para qué", antes de diseñar el "cómo".

Audiencia objetivo

- Define el **perfil del usuario:** edad, intereses, nivel de conocimiento técnico, contexto de uso.
- Ejemplo: "Usuarios entre 18 y 35 años, estudiantes o trabajadores remotos que buscan lugares para conectarse a internet en su ciudad".

Enseñanza clave: diseñar con empatía: ¿Qué espera encontrar el usuario y cómo facilitarle ese camino?

Requisitos técnicos

- Determina lo siguiente:
 - Plataforma (web, móvil)
 - Lenguajes de desarrollo (por ejemplo, HTML, CSS, JavaScript, React, etcétera)
 - Librerías y herramientas (por ejemplo, Figma para diseño, Leaflet para mapas)
 - Requisitos específicos del sistema: geolocalización, conexión a internet, permisos del navegador, etcétera.

Recomendación pedagógica: enseñar cómo desglosar los requisitos funcionales (qué debe hacer) y no funcionales (rendimiento, seguridad, etcétera).

Bocetos de alta fidelidad

- Usa herramientas como Figma, Adobe XD o Sketch.
- Los bocetos deben:
 - Mostrar la estructura visual de las pantallas
 - Incluir botones, menús, tarjetas, iconos, etcétera.
 - Mantener consistencia en estilo, tipografía y color.

Aprendizaje esencial: prototipar permite probar ideas antes de codificarlas. Enseña a iterar sobre el diseño.

Estructura de navegación y jerarquía

- Define lo siguiente:
 - Pantalla inicial
 - Flujo de navegación (menú, enlaces, transiciones)
 - Distribución de contenido según prioridad

Sugerencia de enseñanza: mostrar ejemplos de buenas prácticas (por ejemplo, el patrón de navegación tipo hamburguesa o tab bar para móviles).

Integración de API de geolocalización

- **API de geolocalización del navegador (HTML5):** permite obtener la ubicación del usuario con su permiso.
- **Servicios de mapas:**
 - Google Maps: interfaz robusta, documentación amplia, pero con límites gratuitos.
 - Leaflet: libre y ligero, ideal para proyectos educativos o personales.

Recomendación práctica: hacer pruebas con ambos servicios y documentar ventajas/desventajas.

2. Entrega

Documento de planificación

Debe incluir lo siguiente:

- Descripción general del proyecto
- Propósito y objetivos
- Perfil de usuario
- Requisitos técnicos (API, lenguajes, librerías)
- Consideraciones de usabilidad y accesibilidad

Tip didáctico: fomentar una estructura clara, como si se tratara de un documento de especificaciones para un equipo de desarrollo.

Bocetos o prototipos de alta fidelidad

- Incluir al menos:
 - Pantalla principal
 - Pantalla de resultados geolocalizados
 - Pantalla con detalles del lugar (si aplica)
 - Menú de navegación
- Mostrar interacciones básicas y enlaces funcionales.

Sugerencia docente: evaluar tanto el diseño visual como la lógica detrás de la navegación.

Fase II

Desarrollo del prototipo interactivo con geolocalización y personalización de contenidos

1. Geolocalización

Implementación con la API de geolocalización de HTML5

Objetivo: obtener la latitud y longitud del usuario desde su navegador.

Nota didáctica: enseñar cómo manejar permisos y errores. Muchos navegadores requieren HTTPS y autorización explícita del usuario.

Mostrar ubicación en un mapa interactivo

Con Google Maps:

- Requiere API key

Carga el mapa centrado en la ubicación del usuario

Agrega marcador (marker) para señalar su posición

Con Leaflet (más libre):

- Ligero, de código abierto, sin necesidad de API key
- Fácil de usar con L.map, L.marker y L.tileLayer

Consejo práctico: usar Leaflet en proyectos educativos por su facilidad de uso y menor complejidad de configuración.

Buen ejercicio en clase: mostrar cómo se actualiza el mapa al moverse el usuario o recargar la página.

2. Personalización del contenido

API de puntos de interés (POI)

Google Places API:

- Requiere API key con facturación activada.
- Permite buscar lugares cercanos (cafeterías, bancos, hospitales, etcétera).
- Usa coordenadas y tipo de lugar en la consulta.

Foursquare Places:

- Alternativa gratuita (limitada).

- Ideal para pruebas con categorías más variadas.
- Aprendizaje esencial: interpretar documentación de API externas y trabajar con solicitudes HTTP (fetch, axios).

Interacciones con puntos de interés

- Mostrar marcadores dinámicamente en el mapa.

Al hacer clic:

Mostrar nombre, dirección, calificación y horario.

Opcional: abrir en Google Maps o mostrar detalles personalizados.

Recomendación docente: enseñar cómo mapear datos de la API en objetos JS y representarlos visualmente.

3. Diseño adaptable (Responsive Design)

Uso de CSS media queries

Permiten adaptar la interfaz según el tamaño de pantalla:

Consejo práctico: definir puntos de ruptura comunes: 480px, 768px, 1024px.

Uso de Flexbox o Grid

Flexbox: ideal para diseños en una dimensión (filas o columnas).

Grid: útil cuando se necesita una estructura bidimensional (por ejemplo, tarjetas con info).

Nota didáctica: enseñar cómo se combinan media queries + flexbox/grid para una experiencia fluida.

Fase III

1. Optimización del rendimiento

Mejora de la velocidad de carga

Objetivo: reducir el tiempo que tarda la aplicación en estar disponible para el usuario.

- Imágenes optimizadas:
 - Usa formatos modernos como WebP o AVIF.
 - Redimensiona imágenes antes de subirlas.
 - Comprime imágenes con herramientas como TinyPNG, Squoosh o ImageOptim.
- Minificación de recursos:
 - Minifica CSS, JS y HTML para reducir el tamaño de archivo.
 - Elimina scripts o estilos que no se estén usando.
- Carga diferida (lazy loading):
 - Carga imágenes solo cuando son visibles:

Sugerencia docente: mostrar cómo una imagen de 2 MB puede ralentizar toda la experiencia y cómo optimizarla puede mejorar la percepción de calidad.

Uso de Lighthouse

- Lighthouse es una herramienta integrada en Chrome DevTools que evalúa:
 - Rendimiento
 - Accesibilidad
 - Buenas prácticas
 - SEO
 - Aplicación PWA (si aplica)
- Cómo usarlo:
 1. Abrir DevTools (F12)
 2. Ir a la pestaña "Lighthouse"
 3. Ejecutar el análisis (puede filtrar por dispositivo móvil o desktop)

Enseñanza clave: interpretar el informe y priorizar las mejoras más urgentes, como "Eliminar recursos que bloquean el renderizado".

2. Pruebas de usabilidad

Pruebas con usuarios reales

- Realiza pruebas informales con:
 - Compañeros de clase
 - Familiares o amigos sin experiencia técnica
- Pide que usen la app sin guías y observa lo siguiente:
 - ¿Entienden cómo usarla?
 - ¿Se pierden en la navegación?
 - ¿Cometen errores frecuentes?

Método simple: da una tarea como "encuentra el café más cercano" y observa cómo lo hacen.

Recopila retroalimentación

- Preguntas clave:
 - ¿Qué fue fácil de usar?
 - ¿Qué fue confuso o lento?
 - ¿Qué mejorarías?

Consejo didáctico: enseñar que el diseño debe adaptarse a los usuarios, no al gusto del desarrollador.

3. Accesibilidad

Evaluación con herramientas

- WAVE (<https://wave.webaim.org>)
- axe DevTools (extensión de navegador)

Estas herramientas detectan lo siguiente:

- Falta de texto alternativo (alt)
- Contraste bajo entre texto y fondo
- Problemas con estructura HTML (encabezados mal jerarquizados)
- Elementos no accesibles por teclado

Demostración en clase: mostrar cómo navegar con solo el teclado o un lector de pantalla (como NVDA o VoiceOver).

Mejora según resultados

- Agrega alt en imágenes:
- Mejora el contraste:
 - Usa contrast-ratio.com o las sugerencias de WAVE
- Asegura navegación por teclado:
 - Evita div o span interactivos sin roles ni accesibilidad
 - Usa elementos semánticos (button, nav, main, section, etcétera)

Mensaje importante: La accesibilidad no es opcional; es parte del diseño inclusivo y puede beneficiar a todos.

Fase IV

1. Pruebas de jugabilidad

Objetivo:

Validar que las animaciones funcionen correctamente en el juego real, tanto en jugabilidad como en rendimiento.

Pruebas de jugabilidad

- Explicar que probar no es solo "jugar", sino observar activamente:
 - ¿La animación responde bien al presionar teclas?
 - ¿Se siente fluida y coherente con la acción?
 - ¿Alguna animación rompe la inmersión o es demasiado lenta/rápida?
- Sugerir usar una checklist de prueba:
 - Saltos, ataques, colisiones, ciclos de caminar, animaciones en bucle, transiciones.
- Incluir pruebas con distintos tipos de jugador: (quienes no diseñaron el juego), para obtener perspectivas reales.

Impacto en el rendimiento

- Mostrar cómo una animación mal optimizada (por ejemplo, con demasiados frames o efectos complejos) puede:
 - Bajar la tasa de FPS.
 - Provocar glitches visuales o "lag".
- Enseñar cómo usar el Profiler de Unity para detectar cuellos de botella (CPU, GPU, memoria).
- Aconsejar el uso de LOD (Level of Detail) o simplificación de esqueleto si se detectan problemas.

2. Optimización y exportación

Objetivo:

Asegurar que el juego final con animaciones se ejecute de manera eficiente y esté correctamente ensamblado en Unity.

Exportar de Maya a Unity

- Asegurarse de:
 - Exportar en formato .fbx, seleccionando solo lo necesario (malla + esqueleto + animaciones).
 - Hacer "bake animation" si usaron controladores que Unity no interpreta.
- Recalcar la importancia de nombrar archivos y jerarquías claramente.

Consejo: exportar por capas si el archivo es pesado (personaje, efectos, sonidos separados).

Verificación en Unity

- Una vez importado:
 - Verificar que los clips de animación estén accesibles en el Animator.
 - Comprobar que los efectos visuales y sonoros estén sincronizados.
 - Revisar la escala y la posición de los modelos (frecuente problema).

Optimización general

- Tips de optimización:
 - Reducir la cantidad de huesos innecesarios.
 - Usar compresión de animación en Unity (en la pestaña "Rig" y "Animation").
 - Unificar texturas en atlas (si es 2D) o reducir resolución (si no afecta la calidad visible).
 - Limitar el uso de partículas o efectos en ciclos frecuentes.

3. Presentación del Proyecto Final

Objetivo:

Comunicar el proyecto de forma clara, mostrando el valor creativo, técnico y narrativo.

Preparar la presentación del videojuego

- La presentación debe incluir lo siguiente:
 - Narrativa breve (historia y mundo del juego).
 - Personajes principales y estilo visual.
 - Animaciones clave: ¿cómo se usaron los principios de animación?
 - Demostración jugable (si es posible, en video o tiempo real).
- Fomentar el uso de un reel de animación con los clips principales (idle, caminar, saltar, ataque, etcétera).
- Incluir capturas de pantalla del flujo de trabajo: Maya, Unity, storyboard, etcétera.

Documentación técnica

- Debe explicar:
 - Herramientas usadas y por qué se eligieron.
 - Flujo de trabajo seguido: del boceto a la integración.
 - Retos enfrentados y cómo los resolvieron (problemas de exportación, bugs de animación, optimización).
- Puede entregarse en PDF o como parte de un portafolio visual.



Evidencia

Fase I

1. **Documento de planificación del proyecto**
 - Propósito de la aplicación
 - Objetivos de la interfaz
 - Audiencia objetivo
 - Requisitos técnicos
 - Plan de integración de la API de geolocalización y servicio de mapas
2. **Bocetos de alta fidelidad**
 - Pantallas principales
 - Jerarquía visual
 - Flujo de navegación
3. **Mapa de navegación/arquitectura de la información**
 - Estructura de pantallas y enlaces entre ellas



Fase II

1 Implementación de la geolocalización

- Código funcional para obtener latitud y longitud del usuario
- Visualización de ubicación en el mapa (Google Maps o Leaflet)

2 Integración de contenido personalizado

- Uso de API externa (Google Places, Foursquare, etcétera)
- Visualización de puntos de interés cercanos en el mapa
- Interacción con los puntos (mostrar detalles al hacer clic)

3 Diseño responsive

- Maquetación adaptable usando media queries, flexbox o grid
- Pruebas en diferentes tamaños de pantalla (móvil/tablet/escritorio)

Fase III

1 Informe de rendimiento

- Análisis con Lighthouse (captura de resultados)
- Descripción de acciones tomadas para mejorar rendimiento (optimización de imágenes, carga diferida, etcétera)

2 Informe de pruebas de usabilidad

- Reseña de las pruebas con usuarios reales
- Feedback recopilado y mejoras realizadas

3 Informe de accesibilidad

- Evaluación con WAVE o axe
- Capturas de errores detectados
- Mejoras aplicadas según resultados

Fase IV

Incluye lo siguiente:

- Versión funcional de la aplicación (enlace o repositorio)
- Documentación completa (planificación, pruebas, evaluaciones)
- Capturas o prototipos finales
- Checklist final con cumplimientos técnicos y de diseño