

Reto 1. LINGOverse: tu primer juego en la web



Nivel: 2DAW3

Curso 25-26

1ºevaluación

Índice

Enunciado.....	3
Producto final.....	4
Restricciones técnicas de diseño.....	4
Entregables.....	4
Fecha prevista de entrega.....	4
Criterios evaluación módulo DAW.....	5
Criterios evaluación módulo DWES.....	6
Criterios evaluación módulo DIW.....	7
Criterios evaluación módulo SSII.....	8
Criterios evaluación módulo DWC.....	9

Enunciado




Una empresa de ocio digital quiere lanzar una versión web educativa del juego LINGO, donde los usuarios puedan adivinar palabras y personalizar su experiencia.

El objetivo es crear una aplicación web funcional, que permita jugar, guardar resultados y mostrar un ranking de jugadores. El sistema debe incluir pantallas dinámicas (juego, resultado, ranking) y estar desplegado en un servidor Linux accesible desde un dominio local..

¿En qué consiste Lingo?

Lingo es un juego de palabras similar a Wordle, en el que el jugador adivina una palabra oculta en un número limitado de intentos (normalmente 5).

En cada intento, el jugador introduce una palabra del mismo número de letras y el sistema le ofrece pistas visuales por letra:

-  Letra en la posición correcta.
-  Letra presente pero en otra posición.
-  Letra no incluida en la palabra.

Si la palabra no existe el jugador pierde esa ronda, es decir, es como si no hubiera tecleado nada, no genera pistas. Por otro lado, habrá un tiempo límite para teclear cada palabra.

El jugador utiliza esta información para afinar sus siguientes intentos hasta adivinar la palabra o agotar los intentos.

En esta versión el juego Lingo será personalizable, ajustar aspectos visuales como el tema o el tamaño de letra. Además, el sistema guardará las partidas y mostrará un ranking con los resultados de los jugadores.

Producto final

Una aplicación web que incluya:

- Registro e inicio de sesión
- Juego funcional con sistema de pistas y contador de intentos.
- Pantallas de resultado ("has ganado/has perdido") y ranking.
- Personalización visual (tema, fuente, sonido).
- Base de datos con usuarios, diccionario y partidas.
- Despliegue real en servidor Linux accesible desde lingo.local.
- Documentación técnica completa con capturas y pasos reproducibles.

Restricciones técnicas de diseño

Para este primer reto **no se permite** el uso de frameworks CSS (Bootstrap, Tailwind, Materialize, etc.).

El objetivo es demostrar dominio de las propiedades de CSS3, Flexbox, Grid y media queries de forma nativa.

Entregables

- ☐ Aplicación web funcional y desplegada.
- ☐ Repositorio Git con estructura organizada y commits documentados.
- ☐ Carpeta /docs con:
 - ☐ Presupuesto de un servidor para la aplicación y el software necesario.
 - ☐ Guía de estilo.
 - ☐ Documento pdf con la siguiente información:
 - ☐ Borrador inicial de la aplicación.
 - ☐ Captura de Wireframe de Figma y enlace.
 - ☐ Guía de instalación y despliegue.
 - ☐ Arquitectura y descripción del flujo de la aplicación.
 - ☐ Manual de usuario y evidencias.
 - ☐ Capturas del servidor, DNS, acceso en red y juego funcionando
 - ☐ Video Resumen de la aplicación en funcionamiento (Máx 5')

Fecha prevista de entrega

12 de noviembre

Criterios evaluación módulo DAW

Aspecto evaluado	Descripción	Peso
Configuración del entorno Docker	Se ha creado y configurado correctamente el entorno de contenedores (PHP/Laravel, base de datos, servidor web, etc.). El Dockerfile y el docker-compose.yml son funcionales y coherentes.	20 %
Comunicación entre servicios	Los contenedores se comunican correctamente Las redes y volúmenes están correctamente configurados.	15 %
Configuración del dominio interno	El dominio lingo.local se ha configurado correctamente en el DNS interno permitiendo el acceso a la aplicación desde otros equipos de la red.	10%
Despliegue de la aplicación	La aplicación Lingo se levanta sin errores con docker compose up, es accesible desde el navegador, y las rutas y volúmenes persisten correctamente.	25 %
Documentación técnica del despliegue	Se explica paso a paso cómo construir, levantar y acceder a los contenedores, con capturas y ejemplos de comandos (docker ps, docker exec, docker logs, etc.).	15 %
Control de versiones (Git)	Se utiliza Git de forma continua, con commits frecuentes, mensajes descriptivos y al menos una versión etiquetada (v1.0).	15 %

Criterios evaluación módulo DWES

Aspecto evaluado	Descripción	Peso
Autenticación	Se ha utilizado un sistema de autenticación basado en sesiones.	15%
Enrutamiento	Se han definido las rutas necesarias para atender a las peticiones del cliente.	15%
Controladores	Se han creado los métodos necesarios para dar respuesta a las peticiones del cliente..	30 %
Modelo	Se han creado y utilizado en Laravel modelos ORM para la gestión de la BD.	30 %
Comentarios	Los comentarios del código facilitan la comprensión del mismo.	10 %

Criterios evaluación módulo DIW

Aspecto evaluado	Descripción	Peso
Estructura y semántica HTML5	Las vistas del juego o componentes (inicio, partida, ranking, opciones, etc.) están construidas con etiquetas semánticas y una jerarquía lógica de encabezados. Se valora la claridad y coherencia en la organización del contenido.	20%
Maquetación y disposición (CSS / Flex / Grid)	Se utiliza CSS moderno (Flexbox y/o Grid) para organizar los elementos de la interfaz de manera ordenada, equilibrada y adaptable. No se permiten frameworks externos como Bootstrap o Tailwind	20%
Guía de estilo (obligatoria)	Se presenta una guía de estilo completa que define la identidad visual del proyecto: paleta de colores, tipografía, componentes, espaciado, iconografía y comportamientos visuales. Debe reflejar coherencia con el diseño final..	25%
Coherencia visual y diseño estético	Se aplican principios básicos de accesibilidad: contraste suficiente, etiquetas alternativas, foco visible, tamaños de fuente adecuados y estructura comprensible. La navegación es intuitiva y fluida.	10%
Interactividad y microanimaciones	Se implementan efectos visuales suaves y coherentes con el estilo general (hover, transiciones, feedback visual). No deben distraer al usuario ni romper la consistencia visual.	10 %
Diseño Fint mobile	Se realiza un diseño responsivo priorizando su adaptación a dispositivos móviles utilizando media queries	10%
Creatividad y personalización visual	El proyecto muestra originalidad, buenas decisiones estéticas y aplicación de los conocimientos de diseño adquiridos. Se valora la justificación del estilo elegido y su adecuación al producto en la guía de estilo.	5%

Criterios evaluación módulo SSII

Aspecto evaluado	Descripción	Peso
Análisis de requisitos	Identifica correctamente las necesidades del sistema: tipo de aplicación, número de usuarios concurrentes, servicios necesarios (web, base de datos, DNS, etc.).	10%
Selección de hardware	Elige componentes adecuados (procesador, RAM, almacenamiento, conectividad) en función de los requisitos del proyecto. Las especificaciones son coherentes y realistas	50%
Selección de software y sistema operativo	Justifica la elección del sistema operativo (Linux, Windows Server, etc.) y las herramientas necesarias para desplegar la aplicación (Apache, Nginx, MySQL, Docker...)	25%
Presupuesto y fuentes	Incluye precios actualizados, fuentes citadas (proveedores, enlaces o capturas) y cálculo del coste total (compra o suscripción).	10%
Comparativa (opcional)	Incluye una breve comparativa entre alternativas (local vs cloud, proveedor A vs B) o una estimación del coste mensual/anual	5 %

Criterios evaluación módulo DWC

Aspecto evaluado	Descripción	Peso
Lógica del juego en el cliente	<p>Se gestiona el estado de la partida en el navegador: intentos restantes, posición de cada letra y coloreado en el tablero según aciertos / parciales / fallos. El usuario introduce la palabra usando el teclado en pantalla; cuando completa las 5 letras del intento, el sistema bloquea esa fila, comprueba automáticamente la jugada (incluida la validación en diccionario) y pinta los colores de las letras. Se gestiona que en cada ronda haya 5 palabras máximo.</p> <p>El flujo (ganar / perder / fin de intentos) funciona sin recargar la página. La aplicación funciona como una SPA (Single Page Application): toda la interacción del juego y la navegación se hace en el cliente sin recargar la página completa.</p> <p>El flujo (ganar/perder/fin de intentos), funciona sin recargar la página.</p>	35 %
Comunicación asíncrona con el servidor	Se consumen correctamente las rutas del backend mediante fetch/async/await: obtener la palabra secreta, validar la palabra introducida y consultar/enviar datos del ranking. La interfaz se actualiza con la respuesta recibida.	20 %
Gestión del tiempo e interacción	Se implementa el temporizador por intento/turno. Se bloquea la entrada cuando se agota el tiempo o termina la partida.	20 %
Integración de la autenticación / estado de usuario	El front muestra un formulario de login y envía las credenciales al backend. Responde correctamente al resultado del backend (éxito / error): bloquea el juego si no hay usuario autenticado, muestra quién ha iniciado sesión, permite cerrar sesión. No expone la contraseña en claro.	10 %
Calidad del código JavaScript	El código está estructurado (funciones / módulos), usa nombres claros y evita duplicación innecesaria. Incluye comentarios en las partes clave (estado del juego, comunicación con backend, temporizador, autenticación).	15 %