GUÍA DE EVALUACIÓN ADSO

Proyecto: *Aprende Inglés Profesional – Plataforma Web Interactiva por Niveles*
Ficha: 3171084
Programa: Análisis y Desarrollo de Software (ADSO)
Entrega Final y Exposición

---

1. Portada del Proyecto

Título del Proyecto: *Aprende Inglés Profesional – Sistema Web por Niveles*
Integrantes:

- Juan Camilo Hurtado Sánchez
- Daniel Felipe Acosta
- Juan Sebastian Rojas Guiza
- Elkin Gerardo Gutiérrez Martínez

Instructor: Instructor Gian
Fecha de entrega: 01/12/2025
Ficha: 3171084
URL del Proyecto Desplegado:
file:///C:/Users/X250/Downloads/Proyecto-English%20work.html
Repositorio GitHub: https://github.com/Camilo20099/Proyecto_English_Work.git

---

2. Descripción General del Proyecto

El proyecto **Aprende Inglés Profesional** es una plataforma web interactiva diseñada bajo una arquitectura modular y autosuficiente basada en **HTML5, CSS3 y JavaScript Vanilla**, donde todo el comportamiento del sistema se ejecuta del lado del cliente sin dependencias externas.

La aplicación implementa un **sistema de aprendizaje por niveles**, cada uno con actividades progresivas evaluadas en tiempo real. Estos niveles integran elementos de gamificación, retroalimentación automática y control de flujo que guían el avance del usuario.

El sistema incorpora:

- Un motor de ejercicios dinámico con renderizado automático por nivel.
- Un sistema de **retroalimentación inteligente** que explica la causa del error y penaliza el progreso.
- Un **controlador de progreso** que mide la completitud de cada nivel.
- Un **sistema de bloqueo temporal** basado en conteo de reinicios persistido en localStorage.
- Un **panel de administrador** activado por combinación secreta para la gestión de desbloqueos.
- Una arquitectura escalable que permite añadir nuevos ejercicios, niveles y componentes sin afectar la estructura general.

El objetivo es ofrecer una experiencia de aprendizaje estructurada, técnica y eficiente, optimizada para ambientes académicos o autodidactas.

---

3. Problema Identificado y Justificación

Muchas personas que desean aprender inglés abandonan su proceso debido a falta de estructura, ausencia de seguimiento, actividades poco dinámicas o plataformas confusas.

El problema principal identificado es que **los estudiantes requieren una forma simple, ordenada y motivadora de aprender inglés**, que presente:

- progresión clara,
- retroalimentación inmediata,
- control de errores,
- niveles adaptados al ritmo del usuario,
- recursos visuales y auditivos.

Nuestra solución busca llenar ese vacío mediante un sitio web accesible y fácil de usar, que combina ejercicios interactivos, control de progreso y mecánicas de juego para mantener al usuario comprometido.

---

4. Objetivo General

Crear una plataforma web educativa que permita a los usuarios aprender inglés mediante niveles progresivos, actividades interactivas, controles de avance y retroalimentación automática.

Objetivos Específicos

1. Diseñar una estructura por niveles (A1–B1) alineada con competencias básicas de inglés.

2. Implementar sistemas de progreso, bloqueo y reinicio que incentiven disciplina y constancia.

3. Proporcionar actividades prácticas adaptadas a cada nivel.

4. Garantizar accesibilidad, diseño responsivo y navegación clara.

5. Entregar una documentación completa y una presentación final profesional del proyecto.

---

5. Arquitectura del Sitio y Flujo de Navegación

Mapa del Sitio

- **Inicio**
- **Nivel 1 – A1 Básico**
- **Nivel 2 – A2 Pre-intermedio**
- **Nivel 3 – B1 Intermedio**
- **Pruebas y ejercicios**
- **Pantalla de Bloqueo (intentos fallidos)**
- **Panel de Administrador (desbloqueo)**

Flujo de Usuario

1. Ingresa a la plataforma → ve botones de niveles.
2. Selecciona el nivel → se muestra progreso actual y el botón *“Iniciar Nivel”*.
3. Completa actividades → avanza en la barra de progreso.
4. Si falla repetidamente → pantalla de bloqueo.
5. Un administrador puede desbloquear mediante PIN.
6. Una vez completado un nivel → se habilita el siguiente.

---

6. Diseño UX / UI

Principios aplicados

- Interfaz limpia, minimalista y enfocada en el contenido.
- Colores suaves para evitar fatiga visual.
- Botones grandes y comprensibles para usuarios principiantes.
- Retroalimentación clara: verde = acierto, rojo = error.
- Pantallas con mensajes motivadores.
- Accesibilidad mediante:
    - textos alternativos,
    - navegación por teclado,
    - contraste adecuado,
    - etiquetas semánticas HTML5.

Tipografías

- Títulos: *Poppins*

- Contenido: *Inter*

Paleta de colores

- Azul educativo (#1a73e8)
- Verde retroalimentación positiva (#28a745)
- Rojo advertencia (#dc3545)
- Gris suave (#f2f2f2)

---

7. Desarrollo Técnico

Arquitectura del Código

El sistema sigue una arquitectura basada en componentes funcionales:

- **Módulo de Autenticación Local:**
  Maneja registro, validación e inicio de sesión mediante almacenamiento en el navegador (`localStorage`).

- **Módulo de Gestión de Ejercicios:**
  Genera los componentes interactivos según su tipo:
  - *multiple choice*,
  - *fill-in-the-blank*,
  - *drag & drop*,
  - *matching dinámico*.

- **Módulo de Retroalimentación Inteligente:**
  Implementado mediante la función `mostrarMensajeError()`, la cual genera mensajes explicativos y contextualizados según el tipo de error cometido.

- **Módulo de Control de Progreso:**
  Manejado por las funciones:
  - `updateProgress(level)`

    - `decreaseProgress(level)`
    - `resetLevel(level)`
      Estas controlan el avance numérico, los retrocesos por errores y la recarga del nivel.

- **Módulo de Control de Seguridad:**
  Gestiona el conteo de reinicios e integra:
    - bloqueo automático por exceso de reinicios,
    - pantalla modal de bloqueo,
    - botón administrativo con PIN oculto.

- **Persistencia del Sistema:**
  Todo el estado crítico (reinicios, usuarios, accesos) se almacena en **localStorage**, garantizando persistencia cliente-local sin necesidad de base de datos externa.

Nuevas Funcionalidades Técnicas

Retroalimentación Inteligente

Cada ejercicio, al ser respondido incorrectamente, dispara un mensaje explicando la causa del error.
Ejemplos de mensajes generados:

- *"Seleccionaste un verbo incorrecto. Recuerda que 'are' se usa para plurales."*
- *"El orden sintáctico no coincide. En inglés, primero va el sujeto y luego el verbo."*
- *"La palabra en inglés no coincide con su traducción esperada."*

Este módulo mejora la comprensión conceptual y no solo la memorización.

Penalización Dinámica del Progreso

Al cometer un error, la barra de progreso retrocede aplicando una penalización proporcional:

- `penalty = (100 / totalEjerciciosDelNivel) * 0.8`

Esto evita que el usuario avance por azar y lo obliga a prestar atención.

La función optimizada es:

- `decreaseProgress(level)`: animación + retroceso + corrección de color temporal.

Control de Errores por Actividad

Cada actividad posee un contador interno:

- máximo **3 intentos por ejercicio**,
- deshabilitación automática al exceder intentos,
- marca visual de bloqueo del ejercicio.

Esto se maneja con:

- `div.dataset.attempts`
- `div.dataset.completed`

Sistema de Bloqueo por Reinicios

Cada vez que el usuario reinicia un nivel, se incrementa un contador global:

- `localStorage.getItem("contadorReinicios")`

Cuando alcanza el límite (3 reinicios):

- se oculta toda la interfaz,
- se muestra la pantalla de bloqueo,

- se deshabilitan los niveles.

Características nuevas:

- Bloqueo persistente aunque se recargue la página.
- Botón administrativo oculto mediante combinación **Ctrl + Alt + U**.

Sistema de Administrador

Incluye:

- botón oculto,
- función `desbloquearPagina()` que limpia el contador y restaura la UX,
- validación visual.

Este sistema simula un rol administrativo básico.

Drag & Drop Optimizado

Se implementó una función unificada:

- `checkDragAnswer()`

La cual:

- evalúa orden,
- penaliza errores,
- genera retroalimentación,
- marca el ejercicio como completado.

---

8. Descripción de Funcionalidades

1. Progreso por nivel

Permite visualizar cuánto ha avanzado el usuario. La barra se llena con cada respuesta correcta y se reinicia cuando el usuario falla demasiadas veces.

2. Actividades por dificultad

Cada nivel tiene:

- ejercicios de vocabulario,
- ejercicios de comprensión,
- frases para completar,
- pequeñas pruebas orales o auditivas (si aplica).

3. Bloqueo por intentos fallidos

Tres fallos consecutivos activan bloqueo.
La pantalla muestra un mensaje y deshabilita la interacción.

4. Desbloqueo por administrador

Una contraseña secreta permite al instructor o moderador restablecer el acceso.

5. Reinicio de nivel

Ideal para que el usuario repase antes de intentar avanzar.

6. Retroalimentación Inteligente Automática

Cada vez que el usuario falla, el sistema invoca una función especializada que:

- determina el tipo de actividad,
- compara entrada vs. solución,
- genera un mensaje explicativo,
- aplica una penalización de progreso,
- registra el error a nivel de componente.

Esto contribuye a un aprendizaje didáctico guiado por computadora (*Computer-Assisted Instruction*).

7. Penalización Adaptativa del Progreso

La barra no solo avanza con aciertos, sino que retrocede proporcionalmente con errores:

- retroceso animado,
- degradado visual rojo,
- cálculo técnico dinámico según número de actividades.

8. Control Interno de Intentos por Ejercicio

Cada ejercicio tiene:

- contador individual,
- bloqueo posterior a 3 intentos,
- sombra visual gris para indicar estado inactivo.

---

9. Pruebas y Validación

Pruebas de Integración

Se evaluó:

- funcionamiento de eventos globales (dragdrop, click, keydown),
- consistencia entre niveles,
- persistencia localStorage,
- correcto funcionamiento del sistema administrador.

Pruebas de Estrés

Se probó:

- reiniciar repetidamente niveles,
- spam de botones,
- respuestas incorrectas continuas,
- manipulación del DOM.

El sistema se mantuvo estable.

Pruebas de Usabilidad

Incluyeron:

- retroalimentación automática,
- claridad de mensajes,
- accesibilidad,
- comprensión inferida del error.

---

10. Accesibilidad

Checklist cumplido:
✔ navegación con teclado
✔ etiquetas semánticas HTML5
✔ contraste AA
✔ imágenes con alt
✔ botones grandes
✔ mensajes descriptivos

---

11. Repositorio GitHub

Contiene:

- Código HTML, CSS, JS
- README con instrucciones

- Créditos del equipo

- Licencia abierta

12. Conclusiones del Proyecto

- El sistema por niveles facilita el aprendizaje escalonado.
- La barra de progreso motiva al estudiante.
- El bloqueo automático evita que el usuario avance sin aprender.
- El panel administrador serviría para ambientes educativos reales.
- El proyecto cumple con los objetivos planteados en la guía ADSO.

---

13. Trabajo Futuro

- Agregar audio y pronunciación por nivel.
- Añadir reconocimiento de voz.
- Guardado de progreso permanente con base de datos.
- Sistema de usuarios y ranking.
- Nuevos niveles (B2–C1).

---

14. Referencias

- Material educativo Cambridge
- Recursos de gramática básica (A1–B1)

- WCAG accesibilidad
- Documentación MDN Web Docs

Plan UML

Casos de uso:

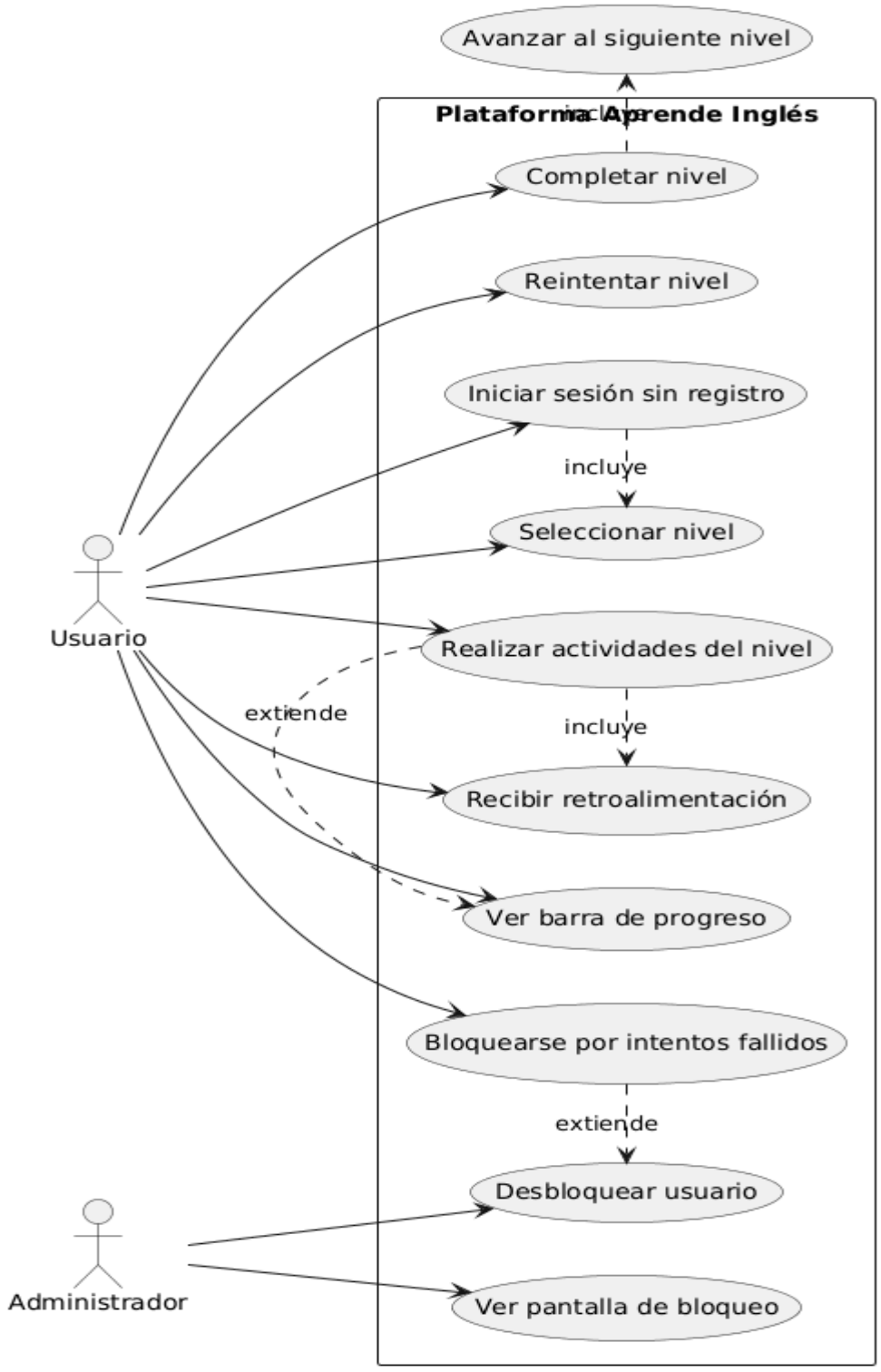


Diagrama de secuencia:

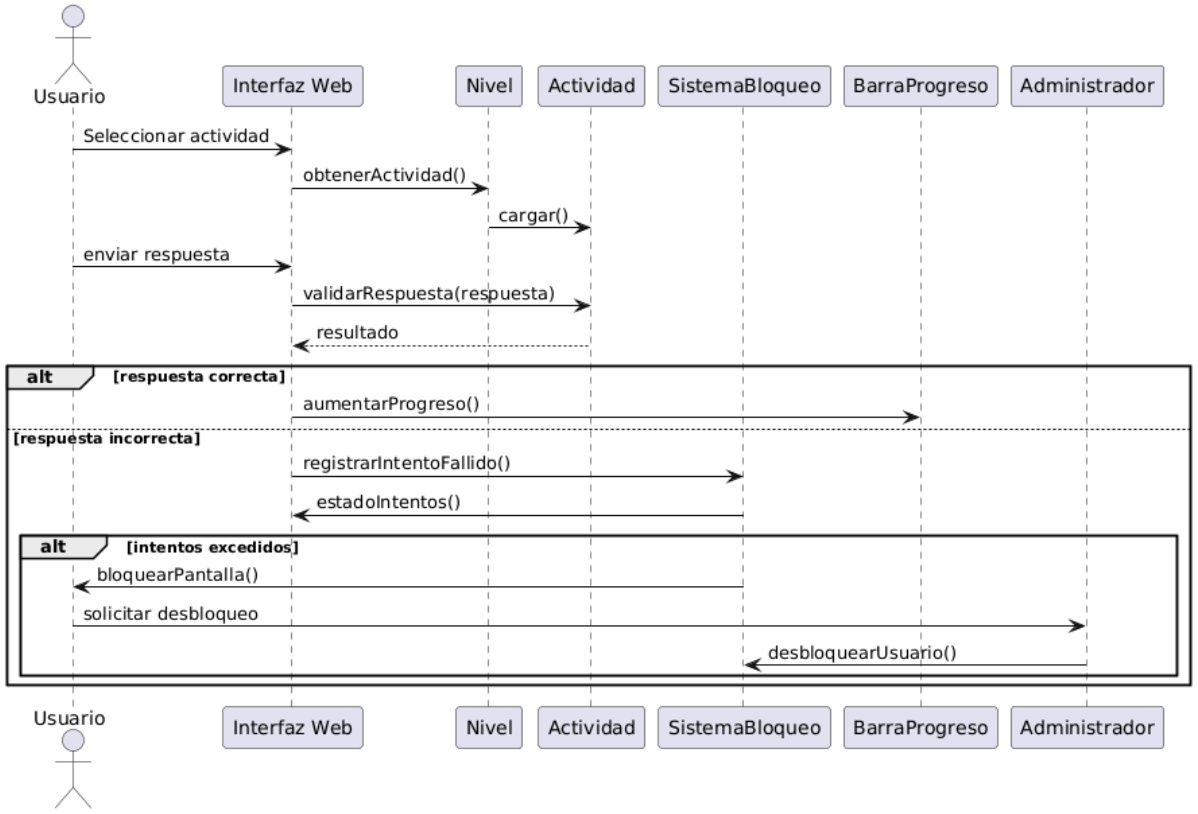


Diagrama de Actividades:

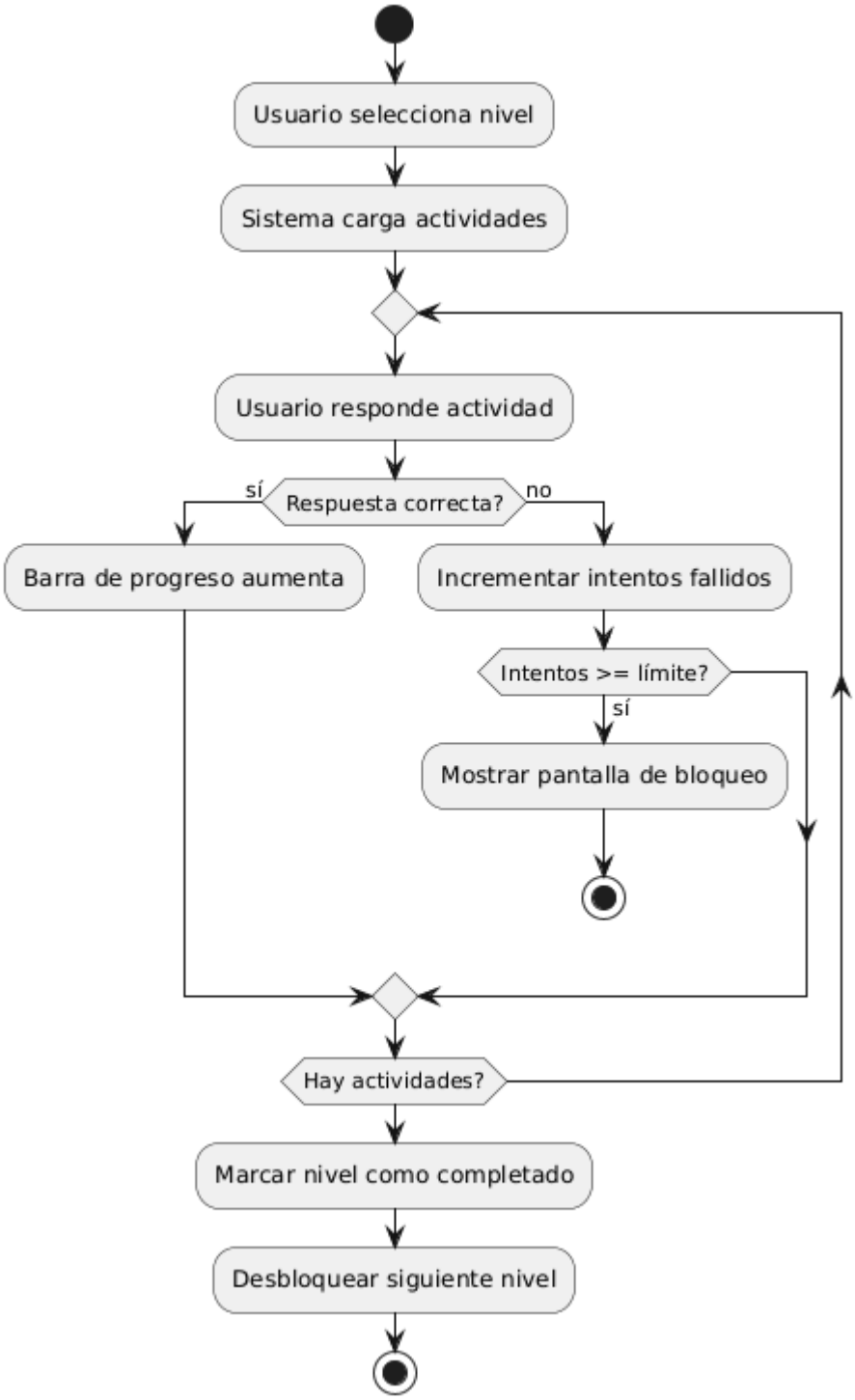
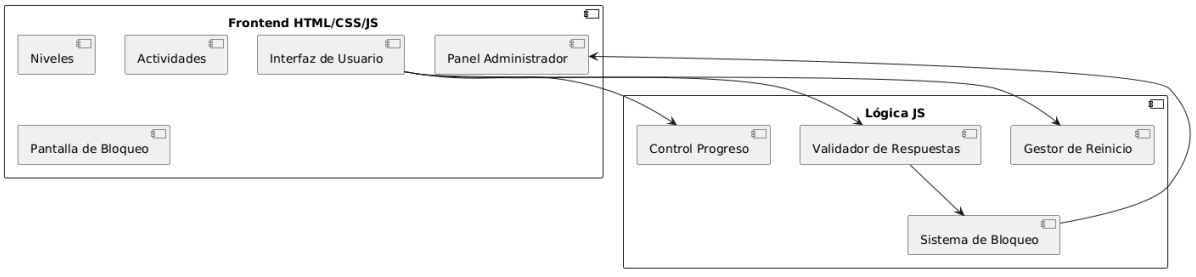


Diagrama de Componentes:



Código explicado:

```
// ================================

// Control de Retroalimentación

// ================================

// Genera un mensaje contextualizado según el error cometido.

// Se inserta dinámicamente dentro del contenedor del ejercicio. function
mostrarMensajeError(div, mensaje) {
    let box = document.createElement("p");
    box.textContent = mensaje; box.style.color =
    "red"; box.style.fontSize = "14px";
    box.style.marginTop = "5px";
    div.appendChild(box);

}



// ================================

// Penalización Dinámica del Progreso

// ================================

// Reduce el porcentaje de la barra de progreso aplicando una penalización proporcional.

// La penalización depende del total de ejercicios del nivel: evita avanzar por azar. function
decreaseProgress(level) {
    const bar = document.getElementById(level + "-progress"); let current
    = parseFloat(bar.style.width) || 0;

    const penalty = (100 / exercisesData[level].length) * 0.8; // penalización adaptativa

    let updated = current - penalty; if
    (updated < 0) updated = 0;

    bar.style.width = updated + "%";

    // Animación visual en rojo para indicar retroceso bar.style.background =
    "linear-gradient(90deg, #ff5252, #ff1744)"; setTimeout(() => bar.style.background
    = "", 500);

}



// ================================

// Validación Drag & Drop

// ================================

// Compara el orden actual de las palabras con el patrón correcto.

// Si falla, penaliza y explica el error; si acierta, marca la actividad como completada. function
checkDragAnswer(div, answer, container, level) {

    let correct = true;

        container.querySelectorAll(".drag-target").forEach((t, i) => { if
            (t.textContent.trim() !== answer[i]) correct = false;

    });

    if (!correct) {
```

```
                mostrarMensajeError(div, "El orden de las palabras no es correcto.");
                decreaseProgress(level);

            } else {

                div.dataset.completed = "true";

            }

            updateProgress(level);

        }



        // ================================

// Reinicio del Nivel



        // ================================

// Restaura los ejercicios, la barra de progreso y deshabilita el botón de completado.

        // También dispara una animación visual para indicar recarga del contenido. function resetLevel(level)
        {
            const container = document.getElementById(level + "-exercises"); const bar =
            document.getElementById(level + "-progress");

            bar.style.width = "0%";                 // reinicio visual loadExercises(level);
                // recargar ejercicios dinámicos document.getElementById(level +
            "-complete").disabled = true;

            // Efecto visual de reinicio
            container.style.opacity = "0";

            setTimeout(() => container.style.opacity = "1", 300);

        }



        // ================================

// Sistema de Bloqueo por Reinicios

        // ================================

// Registra los reinicios y activa el modo "bloqueado" si el usuario abusa del sistema.

        // El estado permanece persistido en localStorage. function
        registrarReinicio() {
            let reinicios = parseInt(localStorage.getItem("contadorReinicios")) || 0; reinicios++;

            localStorage.setItem("contadorReinicios", reinicios);

            if (reinicios >= 3) bloquearPagina();

        }



        // ================================

// Modo Administrador

        // ================================

// Combinación secreta: Ctrl + Alt + U
```

```
// Muestra el botón especial para desbloquear la página. document.addEventListener("keydown", (e) => {
        if (e.ctrlKey && e.altKey && e.key === "u") {
            document.getElementById("admin-unlock-btn").style.display = "block";

    }

});
```