

```
import React, { useState, useEffect, useMemo } from 'react';
import {
  BookOpen,
  Layout,
  CheckCircle,
  Clock,
  BarChart,
  Moon,
  Sun,
  ChevronRight,
  ChevronLeft,
  Award,
  Lightbulb,
  PlayCircle,
  FileText,
  Search,
  ExternalLink
} from 'lucide-react';
```

```
// --- TIPOS Y SCHEMAS ---
```

```
interface Lesson {
  id: string;
  title: string;
  ideaClave: string;
  ejemploAplicado: string;
  actividadPractica: string;
  testRapido: {
    pregunta: string;
    opciones: string[];
    correcta: string;
  }[];
}
```

```
interface Unit {
  id: string;
  title: string;
  summary: string;
  lessons: Lesson[];
}
```

```
interface Course {
  id: string;
  title: string;
  description: string;
  level: string;
  duration: string;
  profile: string;
```

```

objectives: string[];
units: Unit[];
finalEvaluation: { pregunta: string; opciones: string[]; correcta: string; }[];
finalProjects: { title: string; desc: string; }[];
sources: string[];
}

```

// --- PROMPT INTERNO ---

```

const COURSE_BUILDER_PROMPT = (data: any) => `
Actúa como profesor experto y diseñador instruccional senior.
Crea un curso detallado sobre "${data.topic}" para un nivel "${data.membership}"
y un perfil de "${data.profile}".
Objetivo: ${data.objective}. Tiempo: ${data.duration}. Formato: ${data.format}.

```

RESPONDE EXCLUSIVAMENTE EN FORMATO JSON con esta estructura:

```

{
  "title": "Nombre creativo",
  "description": "2-3 frases",
  "level": "${data.membership}",
  "duration": "${data.duration}",
  "profile": "${data.profile}",
  "objectives": ["obj1", "obj2", ...],
  "units": [
    {
      "id": "u1",
      "title": "Nombre unidad",
      "summary": "Resumen claro",
      "lessons": [
        {
          "id": "l1.1",
          "title": "Nombre lección",
          "ideaClave": "4-8 frases didácticas",
          "ejemploAplicado": "Caso real",
          "actividadPractica": "Consigna de acción",
          "testRapido": [{"pregunta": "", "opciones": ["A", "B", "C", "D"], "correcta": "A"}]
        }
      ]
    }
  ],
  "finalEvaluation": [...8-10 preguntas],
  "finalProjects": [{"title": "", "desc": ""}],
  "sources": ["Fuente 1", "Fuente 2"]
}

```

Usa tono humano, cercano, sin mencionar que eres una IA. Idioma: Español de España.

```
`;
```

```

// --- COMPONENTE PRINCIPAL ---
export default function CampusR2R() {
  const [darkMode, setDarkMode] = useState(false);
  const [course, setCourse] = useState<Course | null>(null);
  const [loading, setLoading] = useState(false);
  const [currentLessonId, setCurrentLessonId] = useState<string | null>(null);
  const [completedLessons, setCompletedLessons] = useState<string[]>([]);

  // Formulario
  const [formData, setFormData] = useState({
    topic: "",
    membership: 'standard',
    profile: "",
    objective: "",
    duration: "",
    format: 'Mixto'
  });

  // Persistencia básica
  useEffect(() => {
    const saved = localStorage.getItem('campus_r2r_course');
    if (saved) setCourse(JSON.parse(saved));
    const savedProgress = localStorage.getItem('campus_r2r_progress');
    if (savedProgress) setCompletedLessons(JSON.parse(savedProgress));
  }, []);

  const toggleTheme = () => setDarkMode(!darkMode);

  const generateCourse = async () => {
    setLoading(true);
    // Simulación de llamada a Gemini (En un entorno real, usarías el SDK de Google
    Generative AI)
    // El prompt se enviaría aquí.
    setTimeout(() => {
      const mockCourse: Course = {
        id: '1',
        title: `Maestría en ${formData.topic || 'Diseño Web'}`,
        description: `Un recorrido integral diseñado para que un ${formData.profile}
logre ${formData.objective} en ${formData.duration}.`,
        level: formData.membership,
        duration: formData.duration,
        profile: formData.profile,
        objectives: ["Dominar los fundamentos", "Aplicar técnicas avanzadas",
"Resolver problemas reales"],
        units: [
          {
            id: 'u1',

```

```

    title: 'Fundamentos Esenciales',
    summary: 'Entendiendo la base de todo el ecosistema.',
    lessons: [
      {
        id: 'l1.1',
        title: 'Introducción al concepto',
        ideaClave: 'La clave de esta disciplina reside en la consistencia y la
práctica deliberada.',
        ejemploAplicado: 'Como cuando un arquitecto dibuja planos antes de
poner un solo ladrillo.',
        actividadPractica: 'Escribe en un papel tus 3 dudas principales sobre este
tema.',
        testRapido: [{ pregunta: '¿Cuál es el primer paso?', opciones: ['Planear',
'Ejecutar', 'Dormir', 'Ignorar'], correcta: 'Planear' }]
      },
      {
        id: 'l1.2',
        title: 'Primeros pasos prácticos',
        ideaClave: 'Ahora aplicamos la teoría a un entorno controlado.',
        ejemploAplicado: 'Configuración de tu primer entorno de trabajo.',
        actividadPractica: 'Crea una carpeta de proyecto en tu ordenador.',
        testRapido: [{ pregunta: '¿Qué herramienta usamos?', opciones: ['Editor',
'Martillo', 'Pincel', 'Navegador'], correcta: 'Editor' }]
      }
    ]
  },
  finalEvaluation: [],
  finalProjects: [{ title: 'Proyecto Integrador', desc: 'Desarrolla una solución
completa.' }],
  sources: ['Google Search', 'Documentación Oficial', 'Libros de referencia']
];
setCourse(mockCourse);
setCurrentLessonId(mockCourse.units[0].lessons[0].id);
localStorage.setItem('campus_r2r_course', JSON.stringify(mockCourse));
setLoading(false);
}, 2000);
};

```

```

const markAsCompleted = (id: string) => {
  if (!completedLessons.includes(id)) {
    const newProgress = [...completedLessons, id];
    setCompletedLessons(newProgress);
    localStorage.setItem('campus_r2r_progress', JSON.stringify(newProgress));
  }
};

```

```

const currentLesson = useMemo(() => {
  if (!course) return null;
  for (const unit of course.units) {
    const lesson = unit.lessons.find(l => l.id === currentLessonId);
    if (lesson) return lesson;
  }
  return null;
}, [course, currentLessonId]);

const progress = course
  ? Math.round((completedLessons.length / course.units.reduce((acc, u) => acc +
    u.lessons.length, 0)) * 100)
  : 0;

return (
  <div className={` ${darkMode ? 'bg-slate-900 text-white' : 'bg-slate-50 text-
    slate-900'} min-h-screen font-sans transition-colors duration-300`} >

    {/* HEADER */}
    <header className={` border-b ${darkMode ? 'border-slate-800 bg-slate-900' :
    'border-slate-200 bg-white'} sticky top-0 z-50 p-4`} >
      <div className="max-w-7xl mx-auto flex justify-between items-center">
        <div className="flex items-center gap-2">
          <div className="bg-violet-600 p-2 rounded-lg">
            <BookOpen className="text-white w-6 h-6" />
          </div>
          <h1 className="text-xl font-bold tracking-tight">CAMPUS <span
            className="text-violet-600">R2R</span></h1>
          </div>
          <button onClick={toggleTheme} className="p-2 rounded-full hover:bg-slate-
            200 dark: hover:bg-slate-800 transition-colors">
            {darkMode ? <Sun size={20} /> : <Moon size={20} />}
          </button>
        </div>
      </header>

      {!course ? (
        /* PANTALLA DE DISEÑO (HOME) */
        <main className="max-w-4xl mx-auto py-16 px-4">
          <div className="text-center mb-12">
            <h2 className="text-4xl md:text-5xl font-extrabold mb-4">Crea tu aula
            virtual en minutos</h2>
            <p className="text-lg text-slate-500 dark:text-slate-400">Nuestra IA diseña
            un plan de estudio personalizado con lecciones, actividades y evaluaciones.</p>
          </div>

```

```

<div className={` ${darkMode ? 'bg-slate-800' : 'bg-white'} rounded-2xl
shadow-xl p-8 border ${darkMode ? 'border-slate-700' : 'border-slate-100'}`}>
  <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
    <div className="space-y-2">
      <label className="text-sm font-semibold uppercase tracking-wider
opacity-70">Tema del curso</label>
      <input
        className="w-full p-3 rounded-xl border bg-transparent focus:ring-2
focus:ring-violet-500 outline-none transition-all"
        placeholder="Ej: Química Orgánica, React Avanzado..."
        onChange={(e) => setFormData({...formData, topic: e.target.value})}
      />
    </div>
    <div className="space-y-2">
      <label className="text-sm font-semibold uppercase tracking-wider
opacity-70">Nivel de membresía</label>
      <select
        className="w-full p-3 rounded-xl border bg-transparent focus:ring-2
focus:ring-violet-500 outline-none"
        onChange={(e) => setFormData({...formData, membership:
e.target.value})}
      >
        <option value="standard">Standard</option>
        <option value="plus">Plus</option>
        <option value="premium">Premium</option>
      </select>
    </div>
    <div className="space-y-2">
      <label className="text-sm font-semibold uppercase tracking-wider
opacity-70">Perfil del alumno</label>
      <input
        className="w-full p-3 rounded-xl border bg-transparent focus:ring-2
focus:ring-violet-500 outline-none"
        placeholder="Ej: Estudiante de Bachillerato"
        onChange={(e) => setFormData({...formData, profile: e.target.value})}
      />
    </div>
    <div className="space-y-2">
      <label className="text-sm font-semibold uppercase tracking-wider
opacity-70">Tiempo disponible</label>
      <input
        className="w-full p-3 rounded-xl border bg-transparent focus:ring-2
focus:ring-violet-500 outline-none"
        placeholder="Ej: 4 semanas, 30 min/día"
        onChange={(e) => setFormData({...formData, duration: e.target.value})}
      />
    </div>
  </div>

```

```

</div>

<button
  onClick={generateCourse}
  disabled={loading}
  className="w-full mt-8 bg-violet-600 hover:bg-violet-700 text-white font-
bold py-4 rounded-xl shadow-lg shadow-violet-200 dark:shadow-none transition-
all flex justify-center items-center gap-2"
>
  {loading ? (
    <> <span className="animate-spin mr-2">⦿</span> Diseñando curso...
</>
  ) : (
    <> <Layout size={20} /> Diseñar curso ahora </>
  )}
</button>
</div>
</main>
): (
/* PANTALLA DE AULA VIRTUAL (COURSE VIEW) */
<div className="flex flex-col md:flex-row max-w-[1600px] mx-auto">

  {/* BARRA LATERAL (PLAN DE ESTUDIOS) */}
  <aside className={` w-full md:w-80 h-screen md:sticky top-20 overflow-y-
auto p-6 border-r ${darkMode ? 'border-slate-800' : 'border-slate-200'} `}>
    <h3 className="font-bold text-lg mb-6 flex items-center gap-2">
      <Search size={18} className="text-violet-500" /> Plan de estudios
    </h3>

    <div className="space-y-8">
      {course.units.map((unit) => (
        <div key={unit.id}>
          <h4 className="text-xs font-bold text-slate-400 uppercase mb-3 px-
2">{unit.title}</h4>
          <div className="space-y-1">
            {unit.lessons.map((lesson) => (
              <button
                key={lesson.id}
                onClick={() => setCurrentLessonId(lesson.id)}
                className={` w-full text-left p-3 rounded-lg text-sm transition-all flex
items-center justify-between group ${
                  currentLessonId === lesson.id
                    ? 'bg-violet-600 text-white shadow-md'
                    : 'hover:bg-slate-100 dark:hover:bg-slate-800'
                } `}
              >
                <span className="flex items-center gap-2">

```

```

        {completedLessons.includes(lesson.id) ? (
            <CheckCircle size={16} className={currentLessonId === lesson.id ?
'text-white' : 'text-green-500'} />
        ) : (
            <PlayCircle size={16} className="opacity-50" />
        )}
        {lesson.title}
    </span>
</button>
    )}
</div>
</div>
    )}
</div>
</aside>

```

```

    { /* CONTENIDO PRINCIPAL */ }
    <main className="flex-1 p-6 md:p-12">
        { /* CABECERA DEL CURSO */ }
        <div className={` ${darkMode ? 'bg-slate-800' : 'bg-white'} rounded-3xl p-8
mb-8 border ${darkMode ? 'border-slate-700' : 'border-slate-100'} shadow-sm`} >
            <div className="flex flex-wrap gap-2 mb-4">
                <span className="bg-violet-100 text-violet-700 dark:bg-violet-900
dark:text-violet-200 px-3 py-1 rounded-full text-xs font-bold
uppercase">{course.level}</span>
                <span className="bg-blue-100 text-blue-700 dark:bg-blue-900 dark:text-
blue-200 px-3 py-1 rounded-full text-xs font-bold uppercase flex items-center gap-
1"><Clock size={12}/> {course.duration}</span>
            </div>
            <h2 className="text-3xl font-bold mb-4">{course.title}</h2>
            <p className="text-slate-500 dark:text-slate-400 mb-6 max-w-2xl leading-
relaxed">{course.description}</p>

```

```

            <div className="flex items-center gap-4 border-t pt-6 mt-6 border-slate-
100 dark:border-slate-700">
                <div className="flex-1">
                    <div className="flex justify-between mb-2 text-sm font-medium">
                        <span>Tu progreso</span>
                        <span>{progress}%</span>
                    </div>
                    <div className="w-full bg-slate-200 dark:bg-slate-700 h-2 rounded-full
overflow-hidden">
                        <div className="bg-violet-600 h-full transition-all duration-500" style={{
width: `${progress}%` }}></div>
                    </div>
                </div>
            </div>
        </button>

```



```

        onClick={() => markAsCompleted(currentLessonId!)}
        className="bg-green-500 hover:bg-green-600 text-white px-6 py-2
rounded-xl font-bold transition-all flex items-center gap-2"
      >
        Completar lección
      </button>
    </div>
  </div>

  { /* LECCIÓN ACTUAL */ }
  {currentLesson && (
    <div className="space-y-6">
      <div className="flex items-center gap-3 mb-4">
        <h3 className="text-2xl font-bold">{currentLesson.title}</h3>
      </div>

      { /* BLOQUES DE CONTENIDO */ }
      <div className="grid grid-cols-1 gap-6">
        { /* IDEA CLAVE */ }
        <div className={` ${darkMode ? 'bg-slate-800' : 'bg-white'} p-8 rounded-
2xl border-l-4 border-l-violet-500 shadow-sm`} >
          <div className="flex items-center gap-2 text-violet-500 mb-4 font-bold
uppercase text-xs tracking-widest">
            <Lightbulb size={18} /> Idea Clave
          </div>
          <p className="text-lg leading-relaxed italic opacity-
90">{currentLesson.ideaClave}</p>
        </div>

        { /* EJEMPLO */ }
        <div className={` ${darkMode ? 'bg-slate-800' : 'bg-white'} p-8 rounded-
2xl shadow-sm border ${darkMode ? 'border-slate-700' : 'border-slate-100'}`} >
          <div className="flex items-center gap-2 text-blue-500 mb-4 font-bold
uppercase text-xs tracking-widest">
            <Layout size={18} /> Ejemplo Aplicado
          </div>
          <p className="leading-relaxed">{currentLesson.ejemploAplicado}</p>
        </div>

        { /* ACTIVIDAD */ }
        <div className={` ${darkMode ? 'bg-slate-800' : 'bg-white'} p-8 rounded-
2xl shadow-sm border ${darkMode ? 'border-slate-700' : 'border-slate-100'}`} >
          <div className="flex items-center gap-2 text-orange-500 mb-4 font-
bold uppercase text-xs tracking-widest">
            <FileText size={18} /> Actividad Práctica
          </div>

```

```

        <div className="bg-orange-50 dark:bg-orange-900/20 p-4 rounded-xl
border border-orange-100 dark:border-orange-800">
            {currentLesson.actividadPractica}
        </div>
    </div>

    {/* TEST RÁPIDO */}
    <div className={` ${darkMode ? 'bg-slate-800' : 'bg-white'} p-8 rounded-
2xl shadow-sm border ${darkMode ? 'border-slate-700' : 'border-slate-100'} `}>
        <div className="flex items-center gap-2 text-green-500 mb-4 font-bold
uppercase text-xs tracking-widest">
            <Award size={18} /> Test Rápido
        </div>
        {currentLesson.testRapido.map((test, i) => (
            <div key={i} className="space-y-4">
                <p className="font-bold text-lg">{test.pregunta}</p>
                <div className="grid grid-cols-1 md:grid-cols-2 gap-3">
                    {test.opciones.map(opt => (
                        <button key={opt} className={` p-4 text-left border rounded-xl
hover:border-violet-500 transition-all ${darkMode ? 'border-slate-700 bg-slate-
700/50' : 'border-slate-100 bg-slate-50'} `}>
                            {opt}
                        </button>
                    ))}
                </div>
            </div>
        ))}
    </div>

    {/* NAVEGACIÓN INFERIOR */}
    <div className="flex justify-between items-center py-10">
        <button className="flex items-center gap-2 text-slate-400 hover:text-
slate-600 transition-colors">
            <ChevronLeft /> Lección anterior
        </button>
        <button className="flex items-center gap-2 font-bold text-violet-600
hover:text-violet-700 transition-colors">
            Siguiete lección <ChevronRight />
        </button>
    </div>

    {/* FUENTES */}
    <div className="mt-12 border-t border-slate-200 dark:border-slate-800
pt-8 opacity-60">
        <h5 className="text-sm font-bold uppercase mb-4 flex items-center
gap-2"><ExternalLink size={14}/> Fuentes y Referencias</h5>

```

```
      <ul className="text-xs space-y-2">
        {course.sources.map((s, i) => <li key={i}>{s}</li>)}
      </ul>
    </div>
  </div>
)}
</main>
</div>
)}
</div>
);
}
```