



DATOS DEL ESTUDIANTE

Apellidos y Nombres:	TASAYCO HUACCAMAYTA DIEGO ALONSO	ID:	1552480
Dirección Zonal/CFP:	CFP CHINCHA		
Carrera:	INGENIERIA DE SOFTWARE CON INTELIGENCIA ARTIFICIAL	Semestre:	V
Curso/ Mód. Formativo:	FULLSTACK DEVELOPER SOFTWARE		
Tema de Trabajo Final:	ENTREGABLE 01		

1. INFORMACIÓN**▪ Identifica la problemática del caso práctico propuesto.**

Gestión manual e ineficiente de cursos virtuales que dificulta el registro, organización y consulta de información académica. Falta de un sistema centralizado para administrar la oferta de cursos, su categorización y asignación de docentes.

▪ Identifica propuesta de solución y evidencias.

Desarrollo de un sistema web con base de datos relacional que permita realizar operaciones CRUD sobre cursos virtuales, manteniendo la jerarquía categoría-subcategoría-curso y vinculación con docentes.

- **Respuestas a preguntas guía**

Durante el análisis y estudio del caso práctico, debes obtener las respuestas a las interrogantes:

Pregunta 01:	
Pregunta 02:	
Pregunta 03:	
Pregunta 04:	
Pregunta 05:	

2. PLANIFICACIÓN DEL TRABAJO

▪ Cronograma de actividades:

N°	ACTIVIDADES	CRONOGRAMA					
		13/10	14/10	15/10	16/10	17/10	18/10
1	Crear la base de datos	X					
2	Crear las tablas		X				
3	Insertar los datos iniciales			X			
4	Probar INNER JOIN				X		
5	Hacer el CRUD de curso					X	
6	Revisar y presentar resultados						X

▪ Lista de recursos necesarios:

1. MÁQUINAS Y EQUIPOS	
Descripción	Cantidad
ADM Ryzen 5 5600GT with Radeon Graphics	1
Memoria Ram 16 GB	1
Placa base: B450M PRO-VDH MAX	1

2. HERRAMIENTAS E INSTRUMENTOS	
Descripción	Cantidad
Bloc de notas	1
Word	1
Visual Studio Code	1

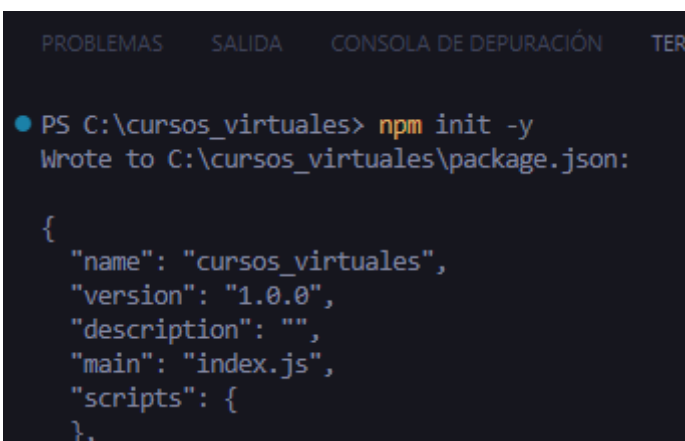
3. MATERIALES E INSUMOS	
Descripción	Cantidad
Git Hub	1

3. DECIDIR PROPUESTA

- Describe la propuesta determinada para la solución del caso práctico

PROPUESTA DE SOLUCIÓN

INICIALIZANDO EL PROYECTO:

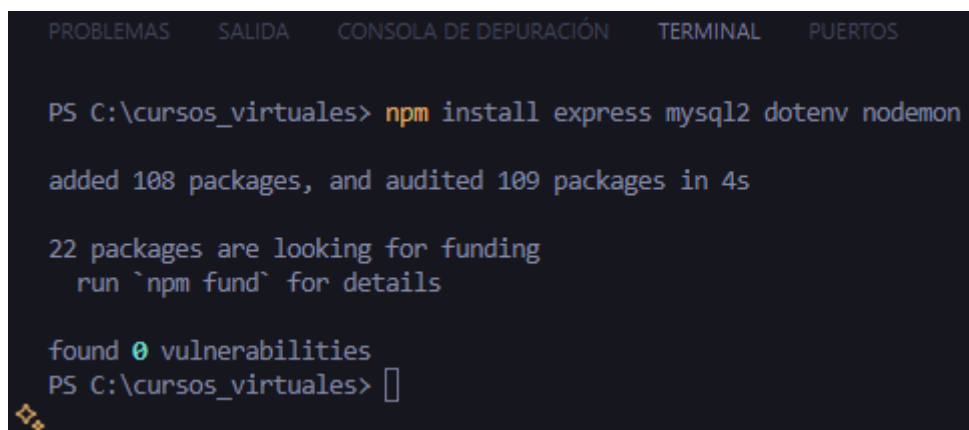


```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\cursos_virtuales> npm init -y
Wrote to C:\cursos_virtuales\package.json:

{
  "name": "cursos_virtuales",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
  },
}
```

INSTALANDO EL FRAMEWORK, ACCESO BD, LEER ARCHIVO CONFIGURACIÓN Y NODEMON PARA EJECUTAR EL SERVIDOR:



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\cursos_virtuales> npm install express mysql2 dotenv nodemon

added 108 packages, and audited 109 packages in 4s

22 packages are looking for funding
  run `npm fund` for details

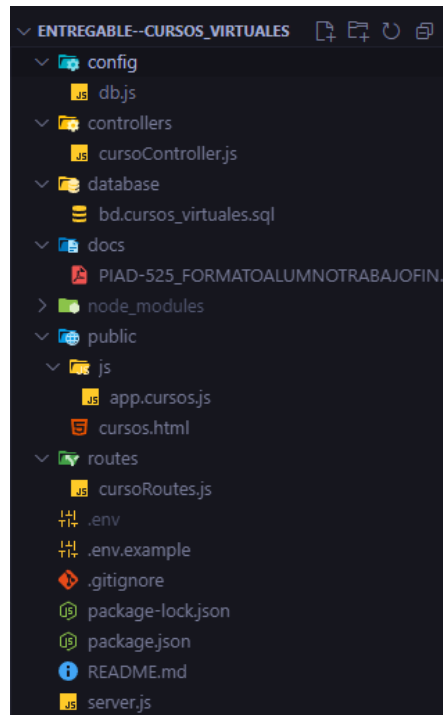
found 0 vulnerabilities
PS C:\cursos_virtuales> 
```

4. EJECUTAR

- Resolver el caso práctico, utilizando como referencia el problema propuesto y las preguntas guía proporcionadas para orientar el desarrollo.
- Fundamentar sus propuestas en los conocimientos adquiridos a lo largo del curso, aplicando lo aprendido en las tareas y operaciones descritas en los contenidos curriculares.

INSTRUCCIONES: Ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. Tomar en cuenta los aspectos de calidad, medio ambiente y SHI.

OPERACIONES / PASOS / SUBPASOS	NORMAS TÉCNICAS - ESTANDARES / SEGURIDAD / MEDIO AMBIENTE
Crear la base de datos cursos_virtuales	
Usar la base creada con USE cursos_virtuales;	
Crear tabla categoria	
Crear tabla subcategoria con FK a categoria	
Crear tabla docente	
Crear tabla curso con FK a subcategoria y docente	
Insertar datos en categoria	
Insertar datos en subcategoria	
Insertar datos en docente	
Insertar datos en curso	
Verificar registros con SELECT * FROM	
Aplicar INNER JOIN para mostrar datos completos	
Probar el CRUD de curso (crear, leer, actualizar, eliminar)	
Comprobar relaciones entre tablas	
Validar los resultados finales	

CARPETAS:**bd.js:**

```
config > db.js > pool
1 //Acceso a datos
2 //Acceder al archivo .env
3 require('dotenv').config()
4
5 const mysql = require('mysql2/promise')
6
7 //Pool de conexiones = acceso
8 const pool = mysql.createPool({
9   host: process.env.DB_HOST,
10  user: process.env.DB_USER,
11  password: process.env.DB_PASSWORD,
12  database: process.env.DB_DATABASE,
13  port: process.env.DB_PORT
14 })
15
16 module.exports = pool
```

cursoController.js

```
controllers > JS cursoController.js > obtenerDocentes > obtenerDocentes
1  const db = require('../config/db')
2
3  > exports.crearCurso = async (req, res) => { ...
23  }
24
25  > exports.obtenerCursos = async (req, res) => { ...
35  }
36
37  > exports.obtenerCursoPorId = async (req, res) => { ...
53  }
54
55  > exports.actualizarCurso = async (req, res) => { ...
77  }
78
79  > exports.eliminarCurso = async (req, res) => { ...
95  }
96
97  > exports.obtenerCategorias = async (req, res) => { ...
107  }
108
109  > exports.obtenerSubcategorias = async (req, res) => { ...
119  }
120
121  > exports.obtenerDocentes = async (req, res) => { ...
131  }
```


app.cursos.js

```
public > js > app.cursos.js > obtenerCursos
1  const API_URL = 'http://localhost:3000/api/cursos'
2
3  const formulario = document.getElementById('form-curso')
4  const tabla = document.querySelector('#tabla-cursos tbody')
5
6  const idcurso = document.getElementById('idcurso')
7  const categoria_id = document.getElementById('categoria_id')
8  const subcategoria_id = document.getElementById('subcategoria_id')
9  const curso = document.getElementById('curso')
10 const docente_id = document.getElementById('docente_id')
11 const fecha_inicio = document.getElementById('fecha_inicio')
12 const fecha_fin = document.getElementById('fecha_fin')
13 const duracion_horas = document.getElementById('duracion_horas')
14 const precio = document.getElementById('precio')
15
16 const btnGuardar = document.getElementById('btnGuardar')
17 const btnCancelar = document.getElementById('btnCancelar')
18
19 let todasSubcategorias = []
20
21 btnCancelar.addEventListener('click', () => {
22   btnGuardar.innerText = 'Guardar'
23   resetearFormulario()
24 })
25
26 > async function cargarCategorias() { ...
40 }
41
42 > categoria_id.addEventListener('change', async (e) => { ...
76 })
77
78 > subcategoria_id.addEventListener('change', (e) => { ...
81 })
82
```

```
82
83 > curso.addEventListener('input', (e) => { ...
92 })
93
94 > async function cargarDocentes() { ...
108 }
109
110 > async function obtenerCursos() { ...
144 }
145
146 > async function eliminarCurso(id, nombreCurso) { ...
164 }
165
166 > async function cargarParaEdicion(cursoItem) { ...
193 }
194
195 > formulario.addEventListener('submit', async (event) => { ...
236 })
237
238 > function resetearFormulario() { ...
247 }
248
249 > document.addEventListener('DOMContentLoaded', () => { ...
253 })
```

cursos.html

```

public > cursos.html > html > body > main.container.my-3
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6    <title>CRUD Cursos</title>
7
8  <link
9    href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
10   rel="stylesheet"
11   integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWUH61DGLwZJEdK2Kadq2F9CUG65"
12   crossorigin="anonymous"
13 />
14 <script
15   src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
16   integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hpcxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
17   crossorigin="anonymous"
18 ></script>
19 </head>
20 <body>
21   <header></header>
22
23   <main class="container my-3">
24     <h3>Cursos Virtuales</h3>
25
26     <form action="" method="post" id="form-curso" autocomplete="off">
27
28       <input type="hidden" id="idcurso">
29
30       <div class="card">
31         <div class="card-header">
32           <strong>Complete los datos solicitados</strong>
33         </div>
34         <div class="card-body">
35           <!-- CATEGORÍA -->
36           <div class="row mb-2">
37             <div class="col-12">
38               <div class="form-floating">
39                 <select
40                   name="categoria_id"
41                   id="categoria_id"

```

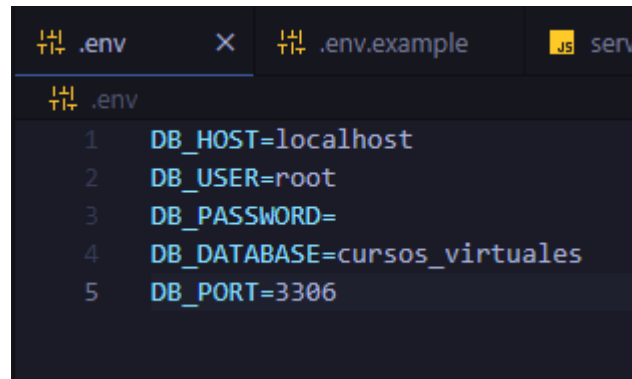
cursoRoutes.js

```
routes > JS cursoRoutes.js > ...
1  const express = require('express')
2  const router = express.Router()
3  const cursoController = require('../controllers/cursoController')
4
5  // Rutas para CRUD de cursos
6  router.post('/', cursoController.crearCurso)
7  router.get('/', cursoController.obtenerCursos)
8  router.get('/:id', cursoController.obtenerCursoPorId)
9  router.put('/:id', cursoController.actualizarCurso)
10 router.delete('/:id', cursoController.eliminarCurso)
11
12 router.get('/categorias/listar', cursoController.obtenerCategorias)
13 router.get('/subcategorias/listar', cursoController.obtenerSubcategorias)
14 router.get('/docentes/listar', cursoController.obtenerDocentes)
15
16 module.exports = router
```

Server.js

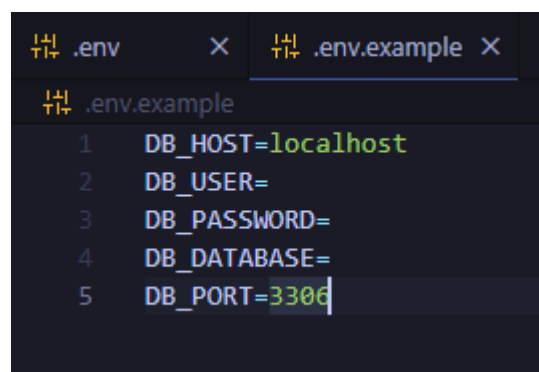
```
JS server.js > ...
1  const express = require('express')
2  const cors = require('cors')
3  const path = require('path')
4
5  const cursoRoutes = require('./routes/cursoRoutes')
6
7  const app = express()
8  const PORT = process.env.PORT || 3000
9
10 app.use(cors({
11   origin: '*',
12   methods: 'GET,HEAD,PUT,PATCH,POST,DELETE',
13   credentials: true
14 }))
15
16 app.use(express.static(path.join(__dirname, 'public')))
17
18 app.get('/', (req, res) => {
19   res.redirect('/cursos.html')
20 })
21
22 app.use(express.json())
23
24 app.use('/api/cursos', cursoRoutes)
25
26 app.listen(PORT, () => {
27   console.log(`Servidor iniciado http://localhost:${PORT}`)
28 })
```

.env



```
.env
1 DB_HOST=localhost
2 DB_USER=root
3 DB_PASSWORD=
4 DB_DATABASE=cursos_virtuales
5 DB_PORT=3306
```

.env.example:



```
.env.example
1 DB_HOST=localhost
2 DB_USER=
3 DB_PASSWORD=
4 DB_DATABASE=
5 DB_PORT=3306
```

EJECUCIÓN

localhost:3000/cursos.html

Cursos Virtuales

Complete los datos solicitados

Categorías Seleccione	
Subcategorías Seleccione	
Descripción del Curso	
Fecha de Inicio del Curso dd/mm/aaaa	Fecha de Fin del Curso dd/mm/aaaa
Docentes Seleccione	
Precio del Curso	Duración del Curso en Horas

Cancelar Guardar

Activar Windows
Ve a Configuración para activar Windows.

localhost:3000/cursos.html

Cursos Virtuales

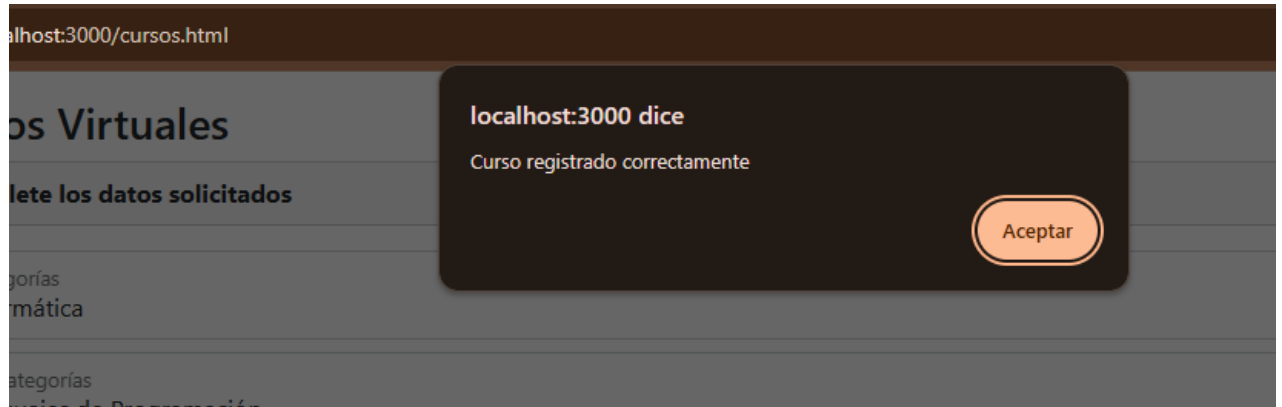
Complete los datos solicitados

Categorías Informática	
Subcategorías Lenguajes de Programación	
Descripción del Curso Php con Laravel	
Fecha de Inicio del Curso 18/10/2025	Fecha de Fin del Curso 25/12/2025
Docentes Luis Rodríguez	
Precio del Curso 350	Duración del Curso en Horas 22

Cancelar Guardar

Activar Windows

SE RISTRÓ CORRECTAMENTE UN NUEVO CURSO:



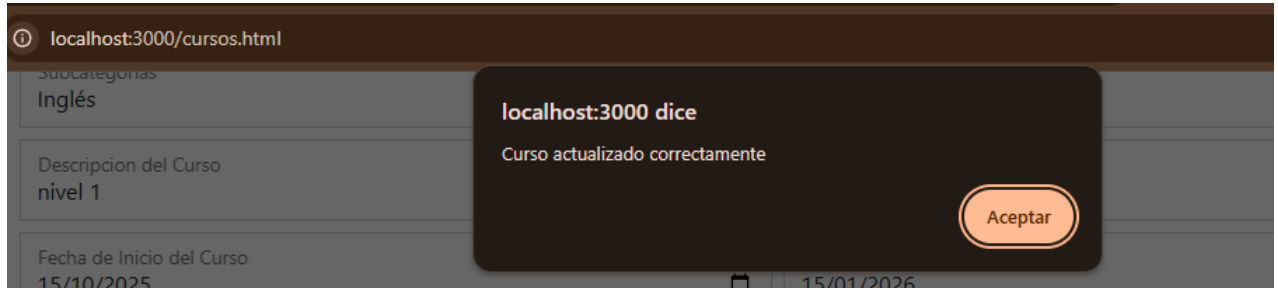
Docentes
 Seleccione

ID	Título del Curso	Categoría	Subcategoría	Docente	Inicio	Fin	Dur. (h)	Precio	Acciones
11	Php con Laravel	Informática	Lenguajes de Programación	Luis Rodríguez	2025-10-18T05:00:00.000Z	2025-12-25T05:00:00.000Z	22 hrs	S/ 350.00	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
8	Java	Informática	Lenguajes de Programación	Juan Pérez	2025-11-08T05:00:00.000Z	2025-12-22T05:00:00.000Z	45 hrs	S/ 259.99	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

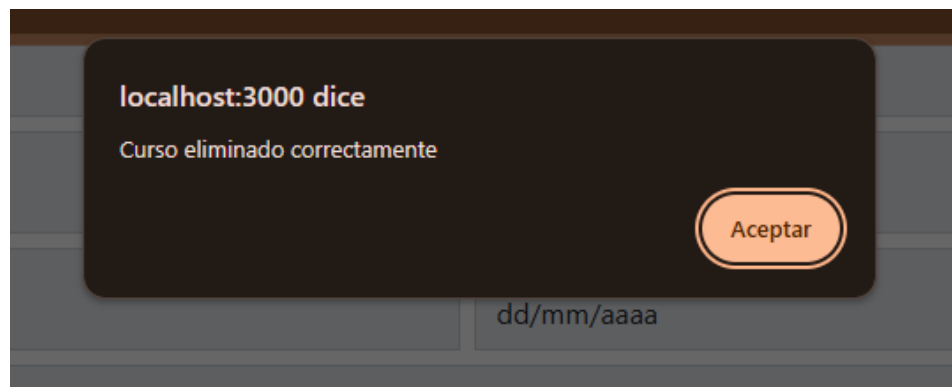
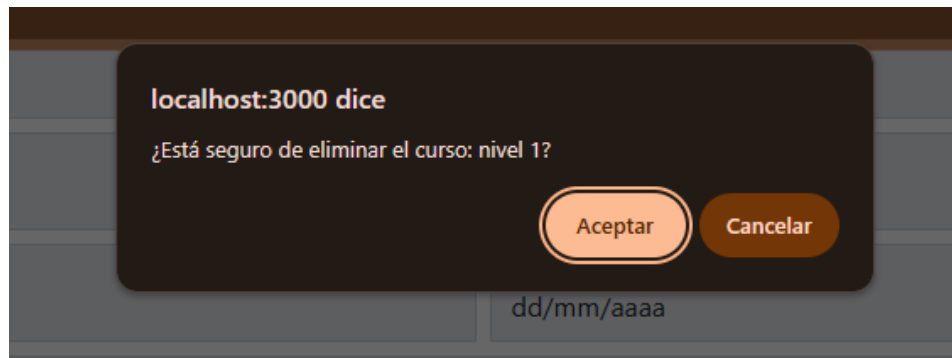
DIBUJO / ESQUEMA / DIAGRAMA DE PROPUESTA

(Adicionar las páginas que sean necesarias)

EDITAR:



ELIMINAR:



GITHUB:

https://github.com/tasaycodiego76-hue/ENTREGABLE--cursos_virtuales.git

 <p>SENATI</p>	ENTREGABLE 01	
	TASAYCO HUACCAMAYTA DIEGO ALONSO	[ESCALA]

5. CONTROLAR

- **Verificar el cumplimiento de los procesos desarrollados en la propuesta de solución del caso práctico.**

EVIDENCIAS	CUMPLE	NO CUMPLE
• ¿Se identificó claramente la problemática del caso práctico?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se desarrolló las condiciones de los requerimientos solicitados?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se formularon respuestas claras y fundamentadas a todas las preguntas guía?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se elaboró un cronograma claro de actividades a ejecutar?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se identificaron y listaron los recursos (máquinas, equipos, herramientas, materiales) necesarios para ejecutar la propuesta?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se ejecutó la propuesta de acuerdo con la planificación y cronograma establecidos?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se describieron todas las operaciones y pasos seguidos para garantizar la correcta ejecución?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se consideran las normativas técnicas, de seguridad y medio ambiente en la propuesta de solución?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
• ¿La propuesta es pertinente con los requerimientos solicitados?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se evaluó la viabilidad de la propuesta para un contexto real?	<input checked="" type="checkbox"/>	<input type="checkbox"/>

6. VALORAR

- Califica el impacto que representa la propuesta de solución ante la situación planteada en el caso práctico.

CRITERIO DE EVALUACIÓN	DESCRIPCIÓN DEL CRITERIO	PUNTUACIÓN MÁXIMA	PUNTAJE CALIFICADO POR EL ESTUDIANTE
Identificación del problema	Claridad en la identificación del problema planteado.	3	
Relevancia de la propuesta de solución	La propuesta responde adecuadamente al problema planteado y es relevante para el contexto del caso práctico.	8	
Viabilidad técnica	La solución es técnicamente factible, tomando en cuenta los recursos y conocimientos disponibles.	6	
Cumplimiento de Normas	La solución cumple con todas las normas técnicas de seguridad, higiene y medio ambiente.	3	
PUNTAJE TOTAL		20	

