



Prueba de Temas 5 y 6. DWES

Tiempo estimado: 3h y 30 minutos

Nombre:

Apellidos:

Fecha:

INSTRUCCIONES:

Lee previamente el enunciado de todas las preguntas.

Realiza los ejercicios en el orden indicado.

Realiza un **commit** cuando finalices cada ejercicio. El mensaje del commit será *ejercicio1*, *ejercicio2*, ...

Si olvidaste algo de un ejercicio previo, puedes realizar un commit más adelante, con el mensaje del ejercicio. Por ejemplo si olvidaste algo del *ejercicio1*, vuelve a hacer commit con mensaje *ejercicio1*

En cada ejercicio se indica el resultado a obtener, sin indicar los detalles concretos, que deberán ser decididos por el alumno/a.

Al finalizar:

- **Sube a la plataforma Moodle un archivo comprimido con el código fuente del proyecto, eliminando previamente las carpetas .next y node_modules.**

NORMAS:

- **SE PERMITE** consultar documentación y los apuntes y actividades realizadas.
- **NO SE PERMITE** el uso de herramientas de IA.

ENUNCIADO DE LA PRUEBA

Realizar una aplicación que permita gestionar información de una **Pizzería**. Crea una **base de datos MySQL local que se llamará pizzeria**.

Necesitaremos gestionar 3 entidades:

REPARTIDOR — 1:N — PEDIDO — N:M — PIZZAS

Un pedido es únicamente entregado por un repartidor.

Los campos o propiedades para la entidad REPARTIDOR serán:

- id
- nombre
- teléfono

Los campos o propiedades para la entidad PEDIDO serán:

- id
- fecha y hora
- nombre cliente
- dirección cliente

Los campos o propiedades para la entidad PIZZA serán:

- id
- nombre
- precio (formato número float)

EJERCICIO1

- Inicia proyecto desde cero.
- Instala los paquetes de prisma que necesites. Si vas a usar iconos o necesitar otros paquetes realiza su instalación también.
- Crea una base de datos mysql vacía llamada **pizzeria**.
- Crea el **esquema de Prisma con los modelos y las relaciones**
- Crea un archivo **lib/prisma.js** con la conexión de prisma.

EJERCICIO2

- Crea un archivo **lib/data.js** con las funciones para recuperar datos.
- Al menos deberás tener 2 funciones por cada entidad:
 - Obtener lista
 - Obtener un ítem
- En caso de necesitar otras funciones de obtención de datos, puedes añadirlas también al archivo.

EJERCICIO3

- En la carpeta **components** crea los componentes para recuperar
 - Lista de repartidores
 - Lista de pedidos
 - Lista de pizzas
- En la carpeta **app** crea las páginas que hagan uso de dichos componentes envueltos en Suspense.

EJERCICIO4

- En la carpeta **components** crea los componentes para recuperar
 - Item de repartidor
 - Item de pedido
 - Item de pizza
- En la carpeta **app** crea las páginas que hagan uso de dichos componentes envueltos en Suspense.

EJERCICIO5

- Crea un archivo **lib/actions.js** con las acciones para mutar datos.
- Deberás tener 3 acciones por cada entidad:
 - Insertar
 - Modificar
 - Eliminar

EJERCICIO6

- En la carpeta **components** crea los componentes para
 - Insertar, Modificar y Eliminar repartidores
 - Insertar, Modificar y Eliminar pedidos
 - Insertar, Modificar y Eliminar pizzas

EJERCICIO7

- Edita el componente Lista de cada entidad para incorporar los componentes anteriores.
- En la carpeta **components** crea un componente Modal (diálogo modal)
- Envuelve los componentes del ejercicio anterior en el componente Modal.

EJERCICIO8

- Para los **pedidos**, edita los componentes de Insertar y Modificar para permitir asignar un repartidor.
- Edita las acciones de servidor correspondientes.

EJERCICIO9

- Para los **pedidos**, edita los componentes de Insertar y Modificar para permitir asociar con pizzas.
- Edita las acciones de servidor correspondientes.

EJERCICIO10

- Convierte los componentes Insertar, Modificar y Eliminar en componentes del lado cliente para la entidad **Pedido**.
- Usa en ellos el hook **useActionState** para gestionar la respuesta del servidor y el estado pending.
- Modifica las acciones para incorporar el argumento prevState.
- Cierra el diálogo modal tras recibir una respuesta de éxito.

CALIFICACIÓN

- Cada ejercicio vale 1 punto como máximo

RÚBRICA PARA CADA APARTADO:

0 puntos	No se ha realizado o se ha realizado de forma completamente errónea.
0,5 puntos	Falta algún resultado requerido en el enunciado.
1 puntos	Se ha realizado y está correcto en su totalidad.