

Historia de Usuario #4: Implementación de gestión de pedidos en tiempo real

Anexo de Documentos Relacionados:

Esta historia de usuario como tal nace de la necesidad de poder optimizar todo lo que tiene que ver con la gestión de pedidos en tiempo real, para que así los que administren el negocio o los cocineros puedan ver y actualizar el estado de los pedidos al momento de una forma eficiente.

Descripción conceptual

Módulo	<i>gestión de pedidos en tiempo real</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Los cocineros o el administrador del restaurante debería poder ver y gestionar los pedidos de los clientes al momento en alguna interfaz la cual le deje actualizar los pedidos con un clic y no complicar mucho la organización operativa.</i>

Backend

URL	Método	Código http
<code>https://api.clickmunch.com/pedidos/gestion</code>	GET	200 400
Caso de uso técnico <i>Cuando se consulta el estado actual de los pedidos si la solicitud que se hace es correcta el sistema devuelve un código 200 y los detalles del pedido, en el caso de que los datos enviados sean incorrectos el sistema va a retornar un código 400 y un mensaje de error</i>		
Datos de entrada 200: <pre>Unset { "status": "success", "data": { "pedido_id": 456 } }</pre>	Datos de salida 200: <pre>Unset { "status": "success", "mensaje": "Estado Preparando" }</pre>	

400: <pre> Unset { "status": "success", "data": { "pedido_id": 000 } } </pre>	400: <pre> Unset { "status": "error", "message": "El ID del pedido no es válido" } </pre>
------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------

URL	Método	Código http
https://api.clickmunch.com/pedidos/gestion	PUT	200 400
Caso de uso técnico <i>Si un administrador actualiza el estado de un pedido el sistema toma la solicitud y devuelve un código 200 si la actualización pudo ser exitosa, en el caso de que la solicitud no cumpla con los parámetros esperados el sistema devuelve un código 400 y un mensaje de error.</i>		
Datos de entrada 200: <div>Unset { "status": "success", "data": { "pedido_id": 456, "estado": "Preparando" } }</div>	Datos de salida 200: <div>Unset { "status": "success", "mensaje": "Estado actualizado exitosamente" }</div>	
400: <div>Unset { "status": "success", "data": { "pedido_id": 456, "estado": "terminad0" } }</div>	400: <div>Unset { "status": "error", "message": "No se pudo actualizar el estado del pedido" }</div>	

--	--

Frontend

- **Interacción esperada:**
 - Los administradores ven una lista de pedidos en tiempo real.
 - Pueden cambiar el estado de cada pedido con un botón de acción.
 - La lista se actualiza dinámicamente con cada cambio.
- **Flujo visual y eventos:**
 - Botón "Actualizar Estado" para cambiar entre "Preparando", "Listo" y "Entregado".
 - Indicador visual para mostrar pedidos nuevos o urgentes.