



Universidad Nacional de Colombia - sede Bogotá  
Facultad de Ingeniería  
Departamento de Sistemas e Industrial  
Ingeniería de Software 1 (2016701)

### Historia de Usuario 1: Registro y autenticación de chazas

#### Anexo de Documentos Relacionados:

##### Base de datos

- Para implementar la siguiente funcionalidad se necesita implementar una nueva en la base de datos llamada “Chazas” donde está integrara al registro y autenticación de chazas

##### Tabla Chaza:

Chaza	
id_chaza	int
id_usuario	int?
nombre	varchar?
descripcion	varchar?
ubicacion	varchar?
foto_id	int?

Medios_pago	
id_medio_pago	int
tipo_pago	varchar?
id_chaza	int?

**Descripción conceptual**

<b>Módulo</b>	<i>Registro y autenticacion de chazas</i>
<b>Descripción de la(s) funcionalidad(es) requerida(s):</b>	<i>Los usuarios tendrán la capacidad de crear "chazas" a su nombre, así como editar la información asociada a estas. Además, podrán eliminar una "chaza" en caso de que no cumpla con las expectativas o no se desarrolle según lo planeado.</i>

**BACKEND**

<b>URL</b>	<b>Método</b>	<b>Código html</b>
/api/chazas	POST	201:: Éxito 400: Solicitud incorrecta, 500: Error interno del servidor
<p align="center"><b>Caso de uso técnico</b></p> <p>Este método permite que un vendedor registre una nueva chaza en la base de datos. Si se proporcionan datos válidos y la operación se realiza con éxito, se devuelve un código 200 con un mensaje de confirmación y los datos de la chaza creada. Si ocurre un error, se devuelve un código 400 o 500 con el mensaje correspondiente.</p>		
<p align="center"><b>Datos de entrada</b></p> <pre>{   "nombre": "La Esquina del Sabor",   "descripcion": "Deliciosas comidas rápidas y bebidas refrescantes.",   "ubicacion": "Bloque A, primer piso",   "foto": "https://ejemplo.com/foto_chaza.jpg",   "id_vendedor": 456,   "horario": "Lunes a Viernes, 8:00 AM - 6:00 PM",   "medios_pago": "Efectivo, Tarjeta, Transferencia" }</pre>	<p align="center"><b>Datos de salida</b></p> <p>200:</p> <pre>{   "status": "success",   "message": "Chaza creada exitosamente",   "data": {     "id_chaza": 789,     "nombre": "La Esquina del Sabor",     "descripcion": "Deliciosas comidas rápidas y bebidas refrescantes.",     "ubicacion": "Bloque A, primer piso",     "foto": "https://ejemplo.com/foto_chaza.jpg",     "id_vendedor": 456,     "horario": "Lunes a Viernes, 8:00 AM -</pre>	

{Authorization : Bearer <token>}	6:00 PM", "medios_pago": "Efectivo, Tarjeta, Transferencia", "puntuacion_promedio": null } }
{}	400: { "status": "error", "message": "Datos inválidos. Por favor verifica los campos requeridos." }
{ "nombre": "La Esquina del Sabor", "descripcion": "Deliciosas comidas rápidas y bebidas refrescantes.", "ubicacion": "Bloque A, primer piso", "foto": "https://ejemplo.com/foto_chaza.jpg", "id_vendedor": 456, "horario": "Lunes a Viernes, 8:00 AM - 6:00 PM", "medios_pago": "Efectivo, Tarjeta, Transferencia" } {Authorization : Bearer <token>}	500: { "status": "error", "message": "No se pudo crear la chaza debido a un error interno." }

URL	Método	Código html
/api/chazas/:idChaza	PUT	200: Éxito, 400: Solicitud incorrecta, 500: Error interno del servidor
<p align="center"><b>Caso de uso técnico</b></p> <p>Este método permite a un vendedor actualizar los datos de una chaza existente en la base de datos. Si el id_chaza es válido y los datos proporcionados son correctos, se actualizan los registros y se devuelve un código 200 con un mensaje de confirmación.</p>		
<p align="center"><b>Datos de entrada</b></p> <pre>{   "nombre": "La Esquina del Sabor 2.0",   "descripcion": "Nueva versión con más opciones de comidas.",   "horario": "Lunes a Viernes, 8:00 AM - 8:00 PM" }</pre> <p>{Authorization : Bearer &lt;token&gt;}</p>	<p align="center"><b>Datos de salida</b></p> <pre>200: {   "status": "success",   "message": "Chaza actualizada exitosamente" }</pre>	
<pre>{ } {Authorization : Bearer &lt;token&gt;}</pre>	<pre>400: {   "status": "error",   "message": "Datos inválidos. Por favor verifica los campos requeridos." }</pre>	
<pre>{   "nombre": "La Esquina del Sabor 2.0",   "descripcion": "Nueva versión con más opciones de comidas.",   "horario": "Lunes a Viernes, 8:00 AM - 8:00 PM" }</pre> <p>{Authorization : Bearer &lt;token&gt;}</p>	<pre>500: {   "status": "error",   "message": "No se pudo actualizar la chaza debido a un error interno." }</pre>	

URL	Método	Código html
/api/chazas/:idChaza	DELETE	200: Éxito, 404:No encontrado, 500: Error interno del servidor
<p align="center"><b>Caso de uso técnico</b></p> <p>Este método permite a un vendedor actualizar los datos de una chaza existente en la base de datos. Si el id_chaza es válido y los datos proporcionados son correctos, se actualizan los registros y se devuelve un código 200 con un mensaje de confirmación.</p>		
Datos de entrada	Datos de salida	
No se requiere {Authorization : Bearer <token>}	200: <pre>{   "status": "success",   "message": "Chaza eliminada exitosamente" }</pre>	
No se requiere {Authorization: Bearer <token>}	404: <pre>{   "status": "error",   "message": "La chaza con el ID proporcionado no existe." }</pre>	
No se requiere {Authorization : Bearer <token>}	500: { <pre>  "status": "error",   "message": "No se pudo eliminar la chaza debido a un error interno." }</pre>	

URL	Método	Código html
/api/chazas/:idChaza	GET	200: Éxito, 400: Solicitud incorrecta, 500: Error interno del servidor
<p align="center"><b>Caso de uso técnico</b></p> <p>Este método permite consultar la información detallada de una chaza existente. Si el id_chaza es válido, se devuelve un código 200 con los datos de la chaza.</p>		
<p align="center"><b>Datos de entrada</b></p> <p>No se requiere {Authorization : Bearer &lt;token&gt;}</p>	<p align="center"><b>Datos de salida</b></p> <p>200:</p> <pre>{   "status": "success",   "data": {     "id_chaza": 789,     "nombre": "La Esquina del Sabor",     "descripcion": "Deliciosas comidas rápidas y bebidas refrescantes.",     "ubicacion": "Bloque A, primer piso",     "foto": "https://ejemplo.com/foto_chaza.jpg",     "id_vendedor": 456,     "horario": "Lunes a Viernes, 8:00 AM - 6:00 PM",     "medios_pago": "Efectivo, Tarjeta, Transferencia",     "puntuacion_promedio": 4.5   } }</pre>	
<p>No se requiere {Authorization : Bearer &lt;token&gt;}</p>	<p>404:</p> <pre>{   "status": "error",   "message": "La chaza con el ID proporcionado no existe." }</pre>	
<p>No se requiere {Authorization : Bearer &lt;token&gt;}</p>	<p>500: {</p> <pre>  "status": "error",   "message": "No se pudo encontrar la chaza debido a un error interno." }</pre>	

## Frontend

### 1. Crear una chaza:

#### Interacción esperada:

El vendedor interactúa con un formulario en la sección de "Mis Chazas" para registrar una nueva chaza.

#### Flujo visual y eventos:

- **Abrir formulario de creación:**
    - El vendedor hace clic en el botón "Agregar Chaza".
    - Aparece un formulario vacío con campos para nombre, descripción, ubicación, foto, horario, y medios de pago.
  - **Registrar los datos:**
    - El vendedor completa los campos obligatorios (ejemplo: nombre, ubicación, etc.).
    - Sube una foto de la chaza, si es necesario.
    - Hace clic en el botón "Guardar".
  - **Solicitud al backend:**
    - Se realiza una solicitud POST al backend con los datos del formulario.
    - Si la operación es exitosa, se agrega la nueva chaza a la lista de chazas del vendedor y la vista se actualiza.
  - **Mensajes:**
    - Éxito: "Chaza creada exitosamente".
    - Error: "No se pudo crear la chaza. Intenta nuevamente más tarde".
- 

### 2. Actualizar una chaza:

#### Interacción esperada:

El vendedor interactúa con un formulario en la página de detalles de su chaza.

#### Flujo visual y eventos:

- **Abrir formulario de edición:**
  - El vendedor hace clic en el botón "Editar".
  - Se despliega un formulario con los campos actuales de la chaza prellenados.
- **Actualizar los datos:**
  - El vendedor edita los campos deseados (por ejemplo, nombre, horario, descripción, etc.).
  - Hace clic en el botón "Guardar cambios".
- **Solicitud al backend:**
  - Se realiza una solicitud PUT al backend con los datos editados.
  - Si la operación es exitosa, se actualiza la vista con los nuevos datos de la chaza.
- **Mensajes:**
  - Éxito: "Chaza actualizada exitosamente".
  - Error: "No se pudo actualizar la chaza. Intenta nuevamente más tarde".

### 3. Eliminar una chaza:

#### Interacción esperada:

El vendedor interactúa con un botón de eliminación en la página de detalles de su chaza.

#### Flujo visual y eventos:

- **Iniciar eliminación:**
    - El vendedor hace clic en el botón rojo "Eliminar".
    - Aparece un cuadro de diálogo de confirmación: "¿Estás seguro de que deseas eliminar esta chaza?".
  - **Confirmar acción:**
    - Si el vendedor confirma, el cuadro de diálogo se cierra y se realiza una solicitud DELETE al backend.
  - **Solicitud al backend:**
    - Si la operación es exitosa, se redirige al vendedor a la lista de sus chazas con un mensaje de confirmación.
  - **Mensajes:**
    - Éxito: "Chaza eliminada exitosamente".
    - Error: "No se pudo eliminar la chaza. Intenta nuevamente más tarde".
- 

### 4. Consultar una chaza:

#### Interacción esperada:

Cualquier usuario interactúa con un enlace o tarjeta en la lista de chazas para ver los detalles de una específica.

#### Flujo visual y eventos:

- **Abrir los detalles de la chaza:**
    - El usuario hace clic en una tarjeta o enlace de una chaza.
    - Se carga una nueva vista o modal con los detalles completos de la chaza.
  - **Visualización de datos:**
    - Se muestran los datos obtenidos del backend: nombre, descripción, ubicación, foto, horario, puntuación promedio, y medios de pago.
  - **Mensajes:**
    - Si ocurre un error en la carga: "No se pudieron cargar los detalles de la chaza. Intenta nuevamente más tarde".
-



## 5. Error general en las operaciones:

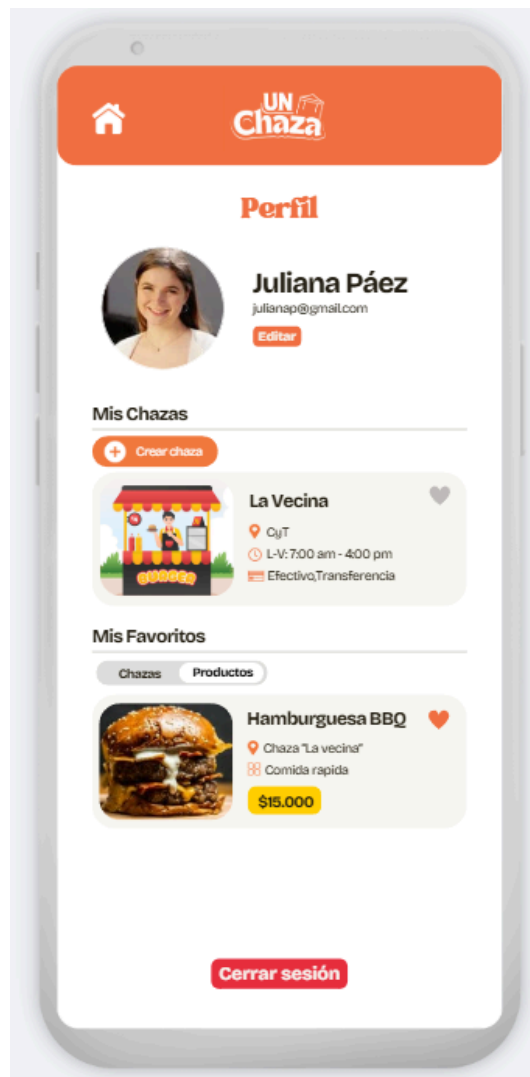
### Flujo visual y eventos:

- Si ocurre un error durante cualquier operación (POST, PUT, DELETE o GET):
  - Mostrar un mensaje de error en la parte superior de la pantalla o en un modal:
    - "Ocurrió un error al procesar tu solicitud. Intenta nuevamente más tarde".

### Mockups/Prototipos:

- **Mockup de Chazas**

Este mockup presenta todas las chazas asociadas al usuario que ha iniciado sesión en la aplicación. Desde aquí, el usuario puede consultar información detallada sobre cada chaza y gestionar los datos que desea mostrar a otros usuarios.



- **Mockup para la Creación de una Chaza**

El propósito de este mockup es representar el proceso de creación de una chaza, tomando como referencia el modelo de base de datos definido para su estructura y funcionamiento.


← **Crear chaza**

**Nombre**

ej: Chaza de Juanito

**Ubicación**


Calle 123, Ciudad Universitaria




**Descripción**

Escribe una breve descripción de tu producto

**Foto**

 Adjuntar foto



**Medios de pago**

☐ Efectivo ☐ Tarjeta débito/ crédito

☐ Nequi ☐ Transferencia bancaria

☐ Daviplata

**Horario**

Lunes

- **Mockup para Eliminar una Chaza**

Este mockup ilustra el proceso para eliminar una chaza asociada al usuario que ha iniciado sesión. Desde esta interfaz, el usuario puede seleccionar la chaza que desea eliminar, confirmando la acción antes de proceder para evitar errores.



**Mockup para Obtener una Chaza por ID**

Este mockup muestra cómo se puede consultar una chaza específica utilizando su ID. La interfaz permite al usuario buscar una chaza en particular y visualizar toda la información detallada relacionada con ella.

