

Documentación de Endpoints para la Aplicación de Cocktails: **MODIFICACIONES RESPECTO AL DOCUMENTO ENTREGADO A RUBÉN EN DICIEMBRE / 2024**

## Pantalla para Registrarse (API propia): **ISSUE AD / PMDM**

**POST /users** (url: 'users', endpoints.register)

**Request Body (json):** **INTERCAMBIO EN EL ORDEN DE NAME Y USERNAME**

```
{
  "Name": "XXX",
  "Username": "XXX",
  "Password": "XXX"
}
```

**Respuestas (json):**

**201 Created:**

```
{
  "token": "XXXX"
}
```

**401 Unauthorized:**

```
{
  "error": "Invalid token."
}
```

## Pantalla para Loguearse (API propia): **ISSUE AD / PMDM**

**POST /session** (url: 'sessions', endpoints.login)

**Request Body (json):**

```
{
  "username": "XXX",
  "password": "XXX"
}
```

**Respuesta (json):**

**201 Created:**

```
{
  "token": "XXXX"
}
```

**401 Unauthorized:**

```
{
  "error": "Invalid token."
}
```

## Pantalla Buscador (API pública de cocktails): **ISSUE PMDM**

El buscador genérico, busca por nombre del cocktail. Devuelve varios:

Buscar por nombre



1. Buscar cocktail por nombre: **GET /search.php?s=nombre\_del\_cocktail**  
<https://www.thecocktaildb.com/api/json/v1/1/search.php?s=margarita>

Debemos pasar un parámetro de búsqueda (s):

s (query string): nombre del cocktail que el usuario quiere buscar

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

#### 400 Bad Request:

```
{
  "error": "No se ha encontrado esa bebida en la base de datos."
}
```

Si queremos más información, pulsamos sobre el cocktail que nos interesa y , como tenemos guardado el id del mismo, nos hace una nueva búsqueda en el API por ese id y nos da toda la información del cocktail:

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "idDrink": "xxx",
      "name": "xxx",
```

```

"category": "xxx",
"alcohol / no alcohol": "xxx",
"tipo de vaso": "xxx",
"instrucciones": "xxx",
"strDrinkThumb ": "xxx.jpg",
"ingrediente1": "xxx",
"ingrediente2": "xxx",
"ingrediente3": "xxx",
"ingrediente4": "xxx",
"ingrediente5": "xxx",
"ingrediente6": "xxx",
"cantadadingrediente1": "xxx",
"cantadadingrediente2": " xxx ",
"cantadadingrediente3": "xxx"
"cantadadingrediente4": "xxx",
"cantadadingrediente5": " xxx ",
"cantadadingrediente6": "xxx"
}
]
}

```

#### 400 Bad Request:

```

{
  "error": "No se ha podido mostrar la lista de ingredientes."
}

```

#### 2. Buscar por ingrediente: **GET /list.php?i=list**

<https://www.thecocktaildb.com/api/json/v1/1/list.php?i=list>

Buscar por ingrediente

#### Respuestas (json):

##### 200 OK:

```

{
  "drinks": [
    {
      "strIngredient1": "Light rum"
    },
    {
      "strIngredient1": "Bourbon"
    },
    {
      "strIngredient1": "Vodka"
    },
    {
      "strIngredient1": "Gin"
    },
  ],
}

```

```
{
  "strIngredient1": "Blended whiskey"
},
{
  "strIngredient1": "Tequila"
}
]
}
```

#### 400 Bad Request:

```
{
  "error": "No se ha podido mostrar la lista de ingredientes."
}
```

Una vez que obtienes el listado de ingredientes, al pinchar sobre uno de los ingredientes, buscará dicho ingrediente por nombre en la API pública: **GET /filter.php?i={nombre\_del\_ingrediente}**

[www.thecocktaildb.com/api/json/v1/1/filter.php?i=Gin](http://www.thecocktaildb.com/api/json/v1/1/filter.php?i=Gin)

Debemos pasar un parámetro de búsqueda (i). Nos devuelve las bebidas que contienen dicho ingrediente.

i (query string): nombre del ingrediente que el usuario quiere buscar

#### Respuestas (json):

##### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

#### 400 Bad Request:

```
{
  "error": "No se ha encontrado ese ingrediente en la base de datos."
}
```

Una vez que tenemos el listado de recetas con el ingrediente especificado, si nos interesa saber más sobre una de las recetas en concreto, al pinchar sobre ella se hará otra búsqueda al endpoint público, esta vez por id (ya que lo hemos obtenido en nuestra búsqueda):

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida:

```
{
  "drinks": [
    {
      "id": "xxx",
      "nombre": "xxx",
      "categoria": "xxx",
      "alcohol / no alcohol": "xxx",
      "tipo de vaso": "xxx",
      "instrucciones": "xxx",
      "imagen": "xxx.jpg",
      "ingrediente1": "xxx",
      "ingrediente2": "xxx",
      "ingrediente3": "xxx",
      "ingrediente4": "xxx",
      "ingrediente5": "xxx",
      "ingrediente6": "xxx",
      "cantidadingrediente1": "xxx",
      "cantidadingrediente2": " xxx ",
      "cantidadingrediente3": "xxx",
      "cantidadingrediente4": "xxx",
      "cantidadingrediente5": " xxx ",
      "cantidadingrediente6": "xxx"
    }
  ]
}
```

3. Buscar bebidas con alcohol o sin alcohol:

**GET /filter.php?a=Alcoholic**



<https://www.thecocktaildb.com/api/json/v1/1/filter.php?a=Alcoholic>

No pasamos parámetro, nos devuelve una lista prefijada de bebidas con alcohol.

### Respuestas (json):

**200 OK:**

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

```

    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}

```

**400 Bad Request:**

```

{
  "error": "Lista de bebidas con alcohol no recibida."
}

```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar bebidas sin alcohol: **GET /filter.php?a=Non\_Alcoholic**



[https://www.thecocktaildb.com/api/json/v1/1/filter.php?a=Non\\_Alcoholic](https://www.thecocktaildb.com/api/json/v1/1/filter.php?a=Non_Alcoholic)

No pasamos parámetro, nos devuelve una lista prefijada de bebidas sin alcohol.

**Respuestas (json):****200 OK:**

```

{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}

```

**400 Bad Request:**

```
{
  "error": "Lista de bebidas sin alcohol no recibida."
}
```

4. Buscar bebidas por categoría: **GET /list.php?c=list**

Nos devuelve una lista con las categorías que tiene la API pública:  
<https://www.thecocktaildb.com/api/json/v1/1/list.php?c=list>

**Respuestas (json):****200 OK:**

```
{
  "drinks": [
    {
      "strCategory": "Cocktail"
    },
    {
      "strCategory": "Ordinary Drink"
    },
    {
      "strCategory": "Punch / Party Drink"
    },
    {
      "strCategory": "Shake"
    },
    {
      "strCategory": "Other / Unknown"
    },
    {
      "strCategory": "Cocoa"
    },
    {
      "strCategory": "Shot"
    },
    {
      "strCategory": "Coffee / Tea"
    },
    {
      "strCategory": "Homemade Liqueur"
    },
    {
      "strCategory": "Beer"
    },
    {
      "strCategory": "Soft Drink"
    }
  ]
}
```

**400 Bad Request:**

```
{
  "error": "Lista de categorías no recibida."
}
```

Ahora, el usuario podrá elegir la categoría que más le guste. Recibirá otro listado de bebidas que están dentro de la categoría elegida. Categorías:

- Cocktail
- Ordinary drink
- Punch / party drink
- Shake
- Other / Unknow
- Cocoa
- Shot
- Coffee / tea
- Homemade Liqueur
- Beer
- Soft drink

Buscar cocktails: **GET /filter.php?c=Cocktail** <https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Cocktail>

No pasamos parámetro, nos devuelve una lista prefijada de bebidas guardadas como cóckteles.

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

#### 400 Bad Request:

```
{
  "error": "Lista de bebidas no encontrada."
}
```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan



todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar bebidas "normales": **GET /filter.php?c=Ordinary\_Drink**  
[https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Ordinary\\_Drink](https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Ordinary_Drink)

No pasamos parámetro, nos devuelve una lista prefijada de bebidas guardadas como ordinarias.

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

#### 400 Bad Request:

```
{
  "error": "Lista de bebidas no encontrada."
}
```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar punch/party drink: **GET /filter.php?c=Punch%20/%20Party%20Drink**  
<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Punch%20/%20Party%20Drink>

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
```

```

    "strDrink": "xxx",
    "strDrinkThumb": "xxx.jpg",
    "idDrink": "xxx"
  },
  {
    "strDrink": "xxx",
    "strDrinkThumb": "xxx.jpg",
    "idDrink": "xxx"
  }
]
}

```

**400 Bad Request:**

```

{
  "error": "Lista de bebidas no encontrada."
}

```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Shake drink: **GET /filter.php?c=Shake**

<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Shake>

**Respuestas (json):****200 OK:**

```

{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}

```

**400 Bad Request:**

```

{

```

```
"error": "Lista de bebidas no encontrada."
}
```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Other / Unknown drink: **GET /filter.php?c=Other%20/%20Unknown**

<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Other%20/%20Unknown>

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

#### 400 Bad Request:

```
{
  "error": "Lista de bebidas no encontrada."
}
```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Cocoa drink: **GET /filter.php?c=Cocoa**

<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Cocoa>

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

#### 400 Bad Request:

```
{
  "error": "Lista de bebidas no encontrada."
}
```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Shot drink: **GET /filter.php?c=Shot**

<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Shot>

#### Respuestas (json):

##### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

```

    }
  ]
}

```

**400 Bad Request:**

```

{
  "error": "Lista de bebidas no encontrada."
}

```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Coffee / Tea drink: **GET /filter.php?c=Coffee%20/%20Tea**

<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Coffee%20/%20Tea>

**Respuestas (json):****200 OK:**

```

{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}

```

**400 Bad Request:**

```

{
  "error": "Lista de bebidas no encontrada."
}

```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Homemade Liqueur drink: **GET /filter.php?c=Homemade%20Liqueur**  
<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Homemade%20Liqueur>

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}
```

#### 400 Bad Request:

```
{
  "error": "Lista de bebidas no encontrada."
}
```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Beer drink: **GET /filter.php?c=Beer**  
<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Beer>

### Respuestas (json):

#### 200 OK:

```
{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
```

```

    "idDrink": "xxx"
  },
  {
    "strDrink": "xxx",
    "strDrinkThumb": "xxx.jpg",
    "idDrink": "xxx"
  }
]
}

```

#### 400 Bad Request:

```

{
  "error": "Lista de bebidas no encontrada."
}

```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

Buscar Soft Drink: **GET /filter.php?c=Soft%20Drink**

<https://www.thecocktaildb.com/api/json/v1/1/filter.php?c=Soft%20Drink>

#### Respuestas (json):

##### 200 OK:

```

{
  "drinks": [
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    },
    {
      "strDrink": "xxx",
      "strDrinkThumb": "xxx.jpg",
      "idDrink": "xxx"
    }
  ]
}

```

#### 400 Bad Request:

```

{
  "error": "Lista de bebidas no encontrada."
}

```

Y haremos lo mismo que en el apartado anterior, al pinchar sobre la bebida que nos interesa, se recuperan todos los datos de la misma porque se le pasa el id:

<https://www.thecocktaildb.com/api/json/v1/1/lookup.php?i=idDrink> → i (query String): pasamos el id (que en realidad es un string) para recuperar todos los datos de la receta.

y ahí ya recibimos todos los datos de la bebida.

## Pantalla para Marcar como Favorito (API propia):

**Método HTTP: PUT /favoritos/id :**

**DESCARTADO** permite al usuario marcar un cocktail como favorito que él mismo ha creado

### Cabeceras:

- SessionToken: XXXX

### Request Body (json): no hay

### PUT /favoritos/cocktaildb/:id

permite al usuario marcar un cocktail como favorito de la api publica cocktaildb

### Cabeceras:

- SessionToken: XXXX

### Request Body (json):

```
{
  "name": "nombre del cocktail", //si es la api pública
  "cocktaildb_image_url": "imagen.jpg" //si es la api pública
}
```

### Respuestas (json): (a ambos PUT)

#### 201 Created:

```
{
  "mensaje": "Cocktail añadido a favoritos."
}
```

#### 400 Bad Request:

```
{
  "error": "Faltan campos."
}
```

#### 401 Unauthorized:

```
{
  "error": "Sesión no válida."
}
```

#### 409 Conflict:

```
{
  "error": "Ya pertenece a favoritos."
}
```



## Pantalla para Obtener Favoritos (API propia): **ISSUE AD / PMDM ???**

**Método HTTP: GET /favoritos:**

permite al usuario recuperar todos los cocktails que ha ido marcando como favoritos

### Cabeceras:

- SessionToken: XXXX

### Respuestas (json):

**200 OK**

```
[
  {
    "name": "Margarita",
    "imagen_url": "imagen.jpg",
    "cocktaildb_id": "XXXXX //si es de la api pública
    "customapi_id": "XXXX //si es la api propia - DESCARTADO"
  }
]
```

**401 Unauthorized:**

```
{
  "error": "Invalid session token."
}
```

## Pantalla de Crear Cocktail (API propia): **ISSUE AD / PMDP**

**Método HTTP: POST /cocktail:**

Este endpoint permite crear un nuevo cóctel, almacenando su nombre, instrucciones, ingredientes, tipo de vaso y una imagen en formato Base64.

### Cabeceras:

- SessionToken: abc

### Request Body (json):

```
{
  "name": "nombre del cocktail",
  "instructions": "instrucciones para la elaboración del cocktail.",
  "glass": "tipo de vaso",
  "ingredients": [
    { "name": "Tequila", "quantity": "50ml" },
    { "name": "Triple sec", "quantity": "20ml" },
    { "name": "Lime juice", "quantity": "15ml" }
  ],
  "image": "imagen convertida a Base64."
}
```

### Respuestas (json):

**201 Created:** El cocktail se creó correctamente.

```
{
  "created": "True"
}
```

```
}
```

**400 Bad Request:** Falta algún campo por rellenar.

```
{
  "error": "Falta algún campo."
}
```

**403 Forbidden:** El encabezado Client-Secret no es válido o no está:

```
{
  "error": "Invalid CLIENT_SECRET header"
}
```

## Pantalla de Ver Cocktails (API pública): ISSUE PMDM

**Método HTTP: GET /random.php**

<https://www.thecocktaildb.com/api/json/v1/1/random.php>

**Respuesta (json):**

**200 OK**

```
{
  "drinks": [
    {
      "strDrink": "Margarita",
      "strDrinkThumb": "https://www.thecocktaildb.com/images/media/drink/5noda61589575158.jpg",
      "idDrink": "11007"
    }
  ]
}
```

## Pantalla de Eliminar Favorito (API propia): ISSUE AD (se implementa en la pantalla de Favoritos)

**DELETE /favorites/:id //para la propia – DESCARTADO**

**Método HTTP: DELETE /favorites/cocktaildb/id //para laapi pública**

**Cabeceras:**

- SessionToken: XXXX

**Respuesta:**

**200 OK**

```
{
  "mensaje": "Favorito eliminado correctamente."
}
```

**404 Not Found**

```
{
  "error": "Favorito no encontrado."
}
```

## **Pantalla de Borrar Cocktail (API propia):** **ISSUE AD** (se implementa en la pantalla de Cocktails)

**Método HTTP: DELETE /cocktail /id:**

Este endpoint permite borrar un nuevo cóctel de la API propia.

### **Cabeceras:**

- SessionToken: XXXX

### **Respuesta:**

#### **200 OK**

```
{  
  "mensaje": "Cocktail eliminado correctamente."  
}
```

#### **404 Not Found**

```
{  
  "error": "Cocktail no encontrado."  
}
```