

Base de datos y endpoints de la aplicación TaquillaWeb

Descripción

TaquillaWeb ofrece a los usuarios registrados la posibilidad de adquirir entradas en diferentes exhibidoras cinematográficas.

Los campos de cada nueva entidad "entrada" deberían quedar almacenados en la tabla tEntradas.

Nombre archivo: [bdtaquillaweb.sql](#)

Tablas

- PK: Primary Key
- FK: Foreign Key
- NN: Not Null
- UQ: Unique
- AI: Auto Increment

Tabla tUsuarios						
Columna	Tipo	PK	FK	NN	UQ	AI
id	int	X		X		X
usuarioNombre	varchar(25)			X	X	
password	varchar(255)			X	X	
email	varchar(50)			X	X	
token	varchar(255)				X	

Tabla tCines						
Columna	Tipo	PK	FK	NN	UQ	AI
id	int	X		X		X
cineNombre	varchar(25)			X	X	
cineLogo	varchar(255)			X		
cantidadSalas	int			X		

Tabla tSalas						
Columna	Tipo	PK	FK	NN	UQ	AI
id	int	X		X		X
numeroSala	int			X		
aforoSala	int			X		
id_cine	int		X	X		
id_pelicula	int		X	X		

PK

FK

Creación de las FK:

Estructura:

tCines_id_cine_tSalas

- **tCines** → la tabla a la que hace referencia
- **id_cine** → el campo de la tabla local (en este caso tSalas)
- **tSalas** → tabla local
- separamos cada parte con guión bajo

Column (tabla local)	Database	Table (tabla referencia)	Column (PK tabla ref.)
id_cine	bdtaquillaweb	tCines	id

tPelículas_id_pelicula_tSalas

- **tPelículas** → la tabla a la que hace referencia
- **id_pelicula** → el campo de la tabla local (en este caso tSalas)
- **tSalas** → tabla local
- separamos cada parte con guión bajo

Column (tabla local)	Database	Table (tabla referencia)	Column (PK tabla ref.)
id_pelicula	bdtaquillaweb	tPelículas	id

Tabla tPelículas						
Columna	Tipo	PK	FK	NN	UQ	AI
id	int	X		X		X
titulo	varchar(25)			X	X	
estreno	boolean			X		
sinopsis	varchar(2000)			X		
peliculaPrecio	double			X		
cartel	varchar(255)			X		

Tabla tEntradas						
Columna	Tipo	PK	FK	NN	UQ	AI
id	int	X		X		X
id_pelicula	int		X	X		
id_usuario	int		X	X		
id_cine	int		X	X		
fecha	date			X		
entradaPrecioUnitario	double			X		
entradaCantButacas	int			X		
entradaPrecioTotal	double			X		

Creación de las FK:

Estructura:

tPelículas_id_película_tEntradas

- **tPelículas** → la tabla a la que hace referencia
- **id_película** → el campo de la tabla local (en este caso tEntradas)
- **tEntradas** → tabla local
- separamos cada parte con guión bajo

Column (tabla local)	Database	Table (tabla referencia)	Column (PK tabla ref.)
id_película	bdtaquillaweb	tPelículas	id

tUsuarios_id_usuario_tEntradas

- **tUsuarios** → la tabla a la que hace referencia
- **id_usuario** → el campo de la tabla local (en este caso tEntradas)
- **tEntradas** → tabla local
- separamos cada parte con guión bajo

Column (tabla local)	Database	Table (tabla referencia)	Column (PK tabla ref.)
id_usuario	bdtaquillaweb	tUsuarios	id

tCines_id_cine_tEntradas

- **tCines** → la tabla a la que hace referencia
- **id_cine** → el campo de la tabla local (en este caso tEntradas)
- **tEntradas** → tabla local
- separamos cada parte con guión bajo

Column (tabla local)	Database	Table (tabla referencia)	Column (PK tabla ref.)
id_cine	bdtaquillaweb	tCines	id

Introducción de datos.

Tabla tUsuarios				
id	usuarioNombre	password	email	token

Los usuarios se crearán posteriormente.

.....

Tabla tCines			
id	cineNombre	cantidadSalas	cineLogo
1	Cines Austin	3	
2	Cines TMA	5	
3	Cines Mix	6	

Cines Logos:

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/logos/logo_Austin.jpg

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/logos/logo_TMA.jpg

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/logos/logo_Mix.jpg

Cines Austin (id=1)	
Hay 4 películas y 3 salas en Cines Austin: <ul style="list-style-type: none">Austin Powers (id=1) en sala 1 (id=1)El Guateque (id=2) en sala 2 (id=2)La Vida de Brian (id=3) en sala 3 (id=3)	Aforo Cines Austin: <ul style="list-style-type: none">sala 1 (id=1) → 50 butacassala 2 (id=2) → 60 butacassala 3 (id=3) → 70 butacas

.....

Cines TMA (id=2)	
Hay 4 películas y 5 salas en Cines TMA: <ul style="list-style-type: none">Blade Runner (id=4) en sala 1 (id=4)Matrix (id=5) en sala 2 (id=5)Tron (id=6) en sala 3 (id=6)2001 A Space Odyssey (id=7) en sala 4 (id=7)Tron (id=6) en sala 5 (id=8)	Aforo cines TMA: <ul style="list-style-type: none">sala 1 (id=4) → 65 butacassala 2 (id=5) → 75 butacassala 3 (id=6) → 90 butacassala 4 (id=7) → 80 butacassala 5 (id=8) → 80 butacas

.....

Cines Mix (id=3)	
Hay 5 películas y 6 salas en Cines Mix: <ul style="list-style-type: none">Blade-Runner (id=4) en sala 1 (id=9)La Vida de Brian (id=3) en sala 2 (id=10)Poltergeist (id=8) en sala 3 (id=11)Regreso al Futuro (id=9) en sala 4 (id=12)Memorias de África (id=10) en sala 5 (id=13)Memorias de África (id=10) en sala 5 (id=13)	Aforo Cines Mix: <ul style="list-style-type: none">sala 1 (id=9) → 70 butacassala 2 (id=10) → 70 butacassala 3 (id=11) → 80 butacassala 4 (id=12) → 80 butacassala 5 (id=13) → 100 butacassala 6 (id=14) → 100 butacas

.....

Tabla tSalas				
Id (AI)	id_cine	id_pelicula	numeroSala	aforoSala
1	1	1	1	50
2	1	2	2	60
3	1	3	3	70
4	2	4	1	65
5	2	5	2	75
6	2	6	3	90
7	2	7	4	80
8	2	6	5	80
9	3	4	1	70
10	3	3	2	70
11	3	8	3	80
12	3	9	4	80
13	3	10	5	100
14	3	10	6	100

.....

Tabla tPeliculas					
id	titulo	estreno	sinopsis	peliculaPrecio	cartel
1	Austin Powers	False		6,5	
2	El Guateque	False		6,5	
3	La Vida de Brian	True		8,5	
4	Blade Runner	False		7,5	
5	Matrix	False		7,5	
6	Tron	True		8,5	
7	2001 A Space Odissey	False		7,5	
8	Poltergeist	False		7,5	
9	Regreso al Futuro	False		7,5	
10	Memorias de Africa	True		8,5	

.....

Carteles y sinopsis:

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/austin_powers.jpg

Primera aventura de Austin Powers (Mike Myers), un peculiar y atractivo espía de los años sesenta, cuyo principal enemigo es el doctor Maligno (interpretado también por Myers). Tras ser ambos sometidos a un proceso de congelación, se despiertan treinta años después en una sociedad completamente distinta a la que conocían. Sin embargo, ellos siguen siendo los mismos.

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/el_guateque.jpg

Hrundi V. Bakshi es un patoso actor de origen hindú que se encuentra rodando una película en el desierto. Por sus continuas meteduras de pata, es despedido del rodaje. Inesperadamente, recibe una invitación para asistir a una sofisticada fiesta organizada por el productor de su última película. Gracias a Hrundi, en la fiesta se producirán las situaciones más disparatadas.

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/la_vida_de_Brian.jpg

Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de desgraciados y tronchantes equívocos le harán llevar una vida paralela a la del verdadero Hijo de Dios. Sus pocas luces y el ambiente de decadencia y caos absoluto en que se haya sumergida la Galilea de aquellos días, le harán vivir en manos de su madre, de una feminista revolucionaria y del mismísimo Poncio Pilatos, su propia versión del calvario.

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/blade_runner.jpg

Rick Deckard (Harrison Ford) es un blade runner, un agente de policía destinado al retiro de replicantes ilegales. Su misión es dar caza a un grupo de cuatro de estos androides, sofisticados NEXUS 6 superiores en fuerza e inteligencia a los humanos, pero diseñados para vivir una corta existencia de cuatro años.

<https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/matrix.jpg>

Thomas Anderson es un brillante programador de una respetable compañía de software. Pero fuera del trabajo es Neo, un hacker que un día recibe una misteriosa visita...

<https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/tron.jpg>

Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición de su padre y se adentra en un mundo digital distinto al original y creado por su padre, de feroces programas y juegos de arcade, y donde su padre ha estado atrapado durante 20 años.

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/2001_a_space_odyssey.jpg

Hace millones de años, antes de la aparición del "homo sapiens", unos primates descubren un monolito que los conduce a un estadio de inteligencia superior. Millones de años después, otro monolito, enterrado en una luna, despierta el interés de los científicos. Por último, durante una misión de la NASA, HAL 9000, una máquina dotada de inteligencia artificial, se encarga de controlar todos los sistemas de una nave espacial tripulada.

<https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/poltergeist.jpg>

La película fue el primer gran éxito de Spielberg como productor. La trama gira en torno a los inquietantes sucesos que acontecen en la casa de una familia que vive en los suburbios, y en la que se sospecha que se está produciendo el fenómeno conocido como «poltergeist».

https://www.jfernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/regreso_al_futuro.jpg

La cinta transcurre en el año 1985, una época en la que el joven Marty McFly lleva una existencia anónima con su novia Jennifer. Los únicos problemas son su familia en crisis y un director al que le

[https://www.ifernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/memorias de africa.jpg](https://www.ifernandez.colexio-karbo.com/22-23/taquillaweb-img/carteles/memorias%20de%20africa.jpg)

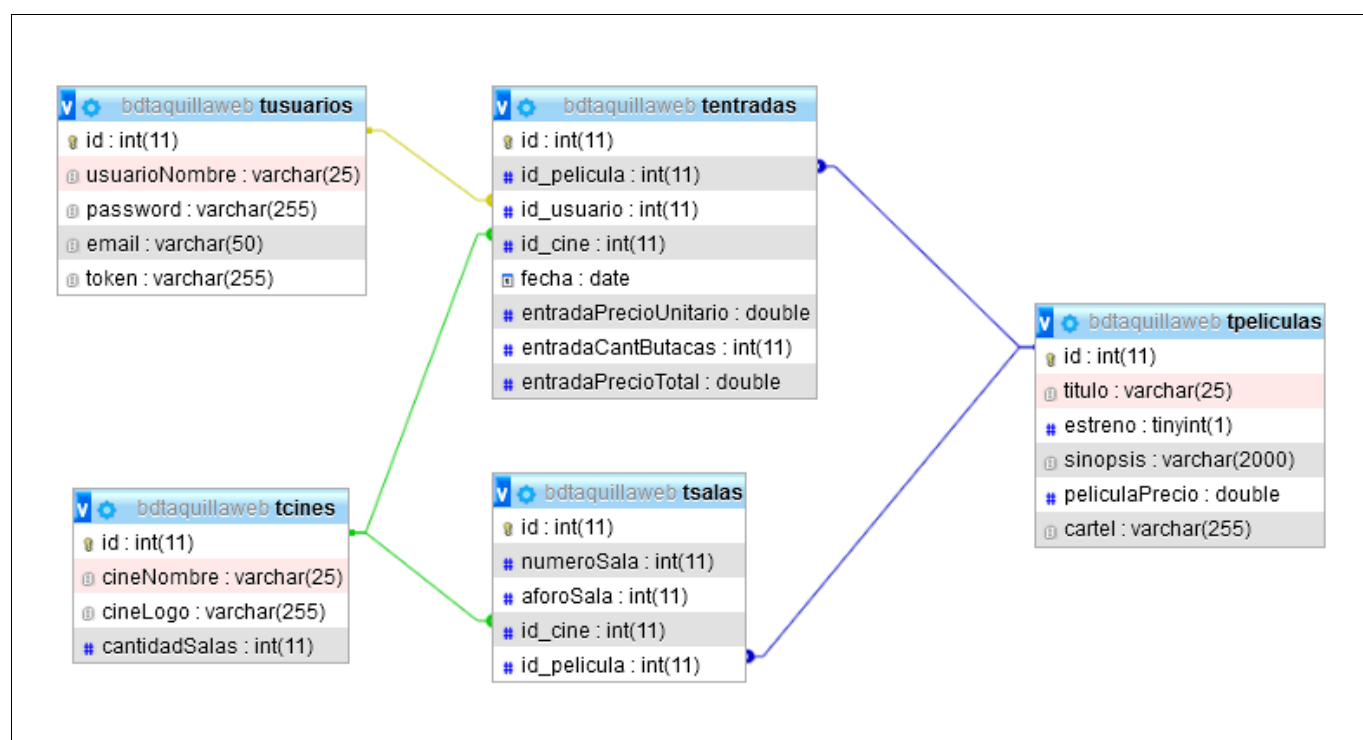
A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.

.....

Tabla tEntradas							
Id (AI)	id_pelicula	id_usuario	id_cine	fecha	entradaNum Butacas	entradaPrecio Unitario	entradaPrecioT otal

Los campos estarán vacíos hasta que haya usuarios que hayan solicitado entradas.

Esquema relacional



Definición de los endpoints de la aplicación.

Cabeceras de una petición GET:

Una petición GET HTTP incluye las siguientes cabeceras opcionales y obligatorias:

- Línea de solicitud: La línea de solicitud es la primera línea de una solicitud HTTP y contiene el método de solicitud (en este caso GET), la URL del recurso solicitado y la versión de HTTP utilizada.
- Host: Esta cabecera obligatoria indica el nombre de dominio del servidor al que se está realizando la solicitud.
- User-Agent: Esta cabecera opcional proporciona información sobre el cliente que realiza la solicitud, como el navegador y la versión utilizados.
- Accept: Esta cabecera opcional indica el tipo de contenido que el cliente puede aceptar. Por ejemplo, si un cliente solo acepta documentos HTML, puede incluir "text/html" en esta cabecera.
- Accept-Language: Esta cabecera opcional indica el idioma preferido del cliente para la respuesta.
- Referer: Esta cabecera opcional indica la URL de la página web que enlaza con la página actual.
- Cookie: Esta cabecera opcional se utiliza para enviar cookies al servidor.
- Authorization: Esta cabecera opcional se utiliza para autenticar al cliente en el servidor mediante un esquema de autenticación determinado.

Estas son algunas de las cabeceras más comunes en una petición GET HTTP, pero hay muchas otras cabeceras que también se pueden incluir dependiendo del caso de uso.

Una petición GET es enviada al servidor...

- *cuando el usuario escribe en la barra de direcciones una URL.*
- *cuando el usuario hace click en un enlace href.*

Cabeceras de una petición POST:

Una petición POST HTTP incluye las siguientes cabeceras opcionales y obligatorias:

- Línea de solicitud: La línea de solicitud es la primera línea de una solicitud HTTP y contiene el método de solicitud (en este caso POST), la URL del recurso solicitado y la versión de HTTP utilizada.
- Host: Esta cabecera obligatoria indica el nombre de dominio del servidor al que se está realizando la solicitud.
- User-Agent: Esta cabecera opcional proporciona información sobre el cliente que realiza la solicitud, como el navegador y la versión utilizados.
- Accept: Esta cabecera opcional indica el tipo de contenido que el cliente puede aceptar. Por ejemplo, si un cliente solo acepta documentos HTML, puede incluir "text/html" en esta cabecera.
- Accept-Language: Esta cabecera opcional indica el idioma preferido del cliente para la respuesta.
- Referer: Esta cabecera opcional indica la URL de la página web que enlaza con la página actual.
- Cookie: Esta cabecera opcional se utiliza para enviar cookies al servidor.
- Content-Type: Esta cabecera obligatoria indica el tipo de contenido del cuerpo de la solicitud. En el caso de una solicitud POST, esta cabecera indica el tipo de datos que se están enviando, como application/json o application/x-www-form-urlencoded.
- Content-Length: Esta cabecera obligatoria indica la longitud en bytes del cuerpo de la solicitud.
- Authorization: Esta cabecera opcional se utiliza para autenticar al cliente en el servidor mediante un esquema de autenticación determinado.

Estas son algunas de las cabeceras más comunes en una petición POST HTTP, pero hay muchas otras cabeceras que también se pueden incluir dependiendo del caso de uso.

Una petición POST es enviada al servidor...

- *cuando el usuario acciona un formulario declarado con la etiqueta HTML `<form> method="post"`*

En este caso, los parámetros (pares clave-valor) viajan en el cuerpo de la petición.

También existen herramientas de línea de comandos como cURL que permiten enviar solicitudes POST.

Cabeceras de respuesta de un servidor web

Las cabeceras de respuesta del servidor web proporcionan información adicional sobre la respuesta enviada por el servidor al cliente. Algunas de las cabeceras de respuesta más comunes son:

- **Status-Line:** La línea de estado indica el código de estado HTTP y una descripción textual correspondiente.
- **Content-Type:** Esta cabecera indica el tipo de contenido de la respuesta, como text/html o application/json.
- **Content-Length:** Esta cabecera indica la longitud en bytes del cuerpo de la respuesta.
- **Server:** Esta cabecera indica el software del servidor web que está respondiendo a la solicitud.
- **Date:** Esta cabecera indica la fecha y hora en que se generó la respuesta.
- **Last-Modified:** Esta cabecera indica la fecha y hora en que se modificó por última vez el recurso solicitado.
- **Cache-Control:** Esta cabecera indica las instrucciones de almacenamiento en caché que el cliente debe seguir.
- **Set-Cookie:** Esta cabecera se utiliza para enviar cookies al cliente.
- **Location:** Esta cabecera se utiliza para redirigir al cliente a una nueva ubicación de URL.
- **ETag:** Esta cabecera proporciona una etiqueta de entidad única para el recurso solicitado.

Estas son solo algunas de las cabeceras de respuesta comunes. El servidor web puede incluir muchas otras cabeceras, dependiendo de la naturaleza de la respuesta y los requisitos del cliente.

Códigos de respuesta:

- **1xx:** Informativos
- **2xx:** Éxito
- **3xx:** Redirección
- **4xx:** Error del cliente
- **5xx:** Error del servidor

Definición previa de los endpoints.

<http://localhost:8000/<ruta del recurso>>

Endpoint 1

POST /users			
Descripción: Sirve para crear un nuevo usuario.			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	POST	/users	HTTP/1.1
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición. Cuando la petición pretende “subir” información al servidor (e.g.: método POST ó PUT) normalmente dicha información se codifica aquí	{ "usuarioNombre": "Paco123", (?) opcional "password": "Abc123", "e-mail": "ñaskjdf@aňlskjfd.clkj", (?) <i>opcional, email o nombre deben ir</i> }		

POST /users → Respuesta del servidor			
	Versión de HTTP	Código	Significado
Línea de estado	HTTP/1.1	201 400 409	Created Bad Request - Faltan parámetros o incorrectos Conflict - Ya hay ese e-mail o nombre de usuario
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta			

Endpoint 2

POST /sessions			
Descripción: Login de usuario.			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	POST	/sessions	HTTP/1.1
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición. Cuando la petición pretende “subir” información al servidor (e.g.: método POST ó PUT) normalmente dicha información se codifica aquí	<pre>{ "usuarioNombre": "Paco123", "password": "Abc123" "email": "ñaskjdf@añoaskjfd.clkj", }</pre>		

POST /sessions → Respuesta del servidor			
	Versión de HTTP	Código	Significado
Línea de estado	HTTP/1.1	201 400 401	Created Bad Request - Faltan parámetros o incorrectos Unauthorized - Contraseña incorrecta
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta (Para 201 - Created)	<pre>{ "sessionToken": "ASDFASDFASDFLASKDF" }</pre>		

Endpoint 3

GET /cines			
Descripción: Devuelve información de todos los cines de la base de datos.			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	GET	/cines	HTTP/1.1
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición			

.....

GET /cines → Respuesta del servidor			
Línea de estado	Versión de HTTP	Código	Significado
	HTTP/1.1	200	OK
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta (Para 200 – OK) Array de objetos JSON. Cada objeto tiene los campos de una empresa.	<pre>[{ "id": Y "cineNombre": "Aaaaaaa", "cantidadSalas": X "cineLogo"="/images/x.png" }, { ... }]</pre>		

Endpoint 4

GET /peliculas			
Descripción: Devuelve información de todas las películas de la base de datos.			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	GET	/peliculas	HTTP/1.1
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición			

.....

GET /peliculas → Respuesta del servidor			
Línea de estado	Versión de HTTP	Código	Significado
	HTTP/1.1	200	OK
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta (Para 200 – OK) Array de objetos JSON. Cada objeto tiene los campos de una película.	<pre>[{ "id": Y, "titulo": "Bbbbbb", "estreno": true, "sinopsis": "Lorem ipsum dolor sit amet consectetur adipiscing elit...", "peliculaPrecio": Z, "cartel": "/images/x.png" }, { ... }]</pre>		

Endpoint 5

GET /peliculas/<id>			
Descripción: Devuelve información de la película con el <id> introducido			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	GET	/peliculas/<id>	HTTP/1.1
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición			

.....

GET /peliculas/<id> → Respuesta del servidor			
Línea de estado	Versión de HTTP	Código	Significado
	HTTP/1.1	200 404	OK Not Found
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta (Para 200 – OK) Objeto JSON con los campos de la película con el id solicitado.	<pre>{ "id": Y, "titulo": "Bbbbbb", "estreno": true, "sinopsis": "Lorem ipsum dolor sit amet consectetur adipiscing elit...", "peliculaPrecio": Z, "cartel": "/images/x.png" }</pre>		

Endpoint 6

GET cines/<id>/peliculas			
Descripción: Devuelve información de las películas del cine cuyo id es el introducido.			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	GET	empresas/<id>/peliculas	HTTP/1.1
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición			

.....

GET cines/<id>/peliculas → Respuesta del servidor			
Línea de estado	Versión de HTTP	Código	Significado
	HTTP/1.1	200 404	OK Not Found
0 ó varias “cabeceras” Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta (Para 200 – OK) Array de objetos JSON. Cada objeto tiene los campos de una película de la empresa seleccionada.	<pre>[{ "id": Y, "titulo": "Bbbbbb", "estreno": true ó false, "sinopsis": "Lorem ipsum dolor sit amet consectetur adipiscing elit...", "peliculaPrecio": Z, "cartel": "/images/x.png" }, { ... }]</pre>		

Endpoint 7

GET /peliculas/estrenos			
Descripción: Devuelve las películas que son estrenos			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	GET	/peliculas/estrenos	HTTP/1.1
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición			

.....

GET /peliculas/estrenos → Respuesta del servidor			
Línea de estado	Versión de HTTP	Código	Significado
	HTTP/1.1	200 404	OK Not Found
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta (Para 200 – OK) Array de objetos JSON. Cada objeto tiene los campos de una película que es estreno.	<pre>[{ "id": Y, "titulo": "Bbbbbb", "estreno": true, "sinopsis": "Lorem ipsum dolor sit amet consectetur adipiscing elit...", "peliculaPrecio": Z, "cartel": "/images/x.png" }, { ... }]</pre>		

Endpoint 8

Requiere token

GET /entradas/cine/id			
Descripción: Devuelve información de la película más taquillera de la semana en el cine cuyo id es el introducido.			
Línea de petición	Método	Ruta del recurso	Versión de HTTP
	GET	/entradas/cine/id	HTTP/1.1
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la petición			

GET /entradas/cine/id → Respuesta del servidor			
Línea de estado	Versión de HTTP	Código	Significado
	HTTP/1.1	200	OK
0 ó varias "cabeceras" Son pares clave: valor			
Línea en blanco			
Cuerpo de la respuesta (Para 200 – OK) Objeto JSON con los campos de la película que cumpla los requisitos.	{ "id": Y, "titulo": "Bbbbb", "estreno": true, "sinopsis": "Lorem ipsum dolor sit amet consectetur adipiscing elit...", "peliculaPrecio": Z, "cartel": "/images/x.png" }		

Endpoint 9: comprar entradas

Requiere token

Endpoint 10: Buscador películas

Requiere token