Contrato Uniforme

Para cada recurso debemos definir el contrato uniforme indicando el tipo de media del recurso, su URI, las operaciones que se pueden realizar sobre el recurso y, por último, las transiciones de estado mediante enlaces (HATEOAS, *Hyperlink As The Engine Of The Application State*)*.*

1. ***Recurso Usuario***
2. ***Recurso Película***
3. ***Recurso Alquiler/Streaming***
4. ***Recurso Compra/Descarga***
5. ***Recurso Login***
6. Recurso Usuario
7. Creación de Usuario

* **Restricción de acceso**: Ninguna.
* **URI**: http://api.videostore.com/users
* **Método HTTP**: POST
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + loginid → identificador de *login* del usuario.
  + password → contraseña del usuario.
  + email→ dirección de correo electrónico del usuario.
* **Respuesta**: Información del usuario creado.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.auth\_token+json
* **Código de estado HTTP**: 201 Created
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Obtención del perfil de un usuario

* **Restricción de acceso**: Usuario(Registrado) y Administrador
* **URI**: http://api.videostore.com/users/{id}
* **Método HTTP**: GET
* **Campos del formulario**
  + Token → Parámetro asociado a la cuenta que verifica la autentificación, Token de administrador en el caso del admin
* **Respuesta**: Información del usuario creado.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.user+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Actualización de un usuario

* **Restricción de acceso**: Usuario(Registrado) y Administrador
* **URI**: http://api.videostore.com/users/{id}
* **Método HTTP**: PUT
* **Tipo de medio de la petición**: application/vnd.dsa.videostore.user+json
* **Campos del formulario**
  + loginid → identificador de *login* del usuario.
  + password → contraseña del usuario.
  + email→ dirección de correo electrónico del usuario.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación,Token de administrador en el caso del admin
* **Respuesta**: Verificación de la actualización.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.user+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Eliminación de un usuario

* **Restricción de acceso**: Usuario(Registrado) y Administrador
* **URI**: http://api.videostore.com/users/{id}
* **Método HTTP**: DELETE
* **Campos del formulario**
  + Token → Parámetro asociado a la cuenta que verifica la autentificación, Token de administrador en el caso del admin
* **Respuesta**: Vacia
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Añadir saldo

* **Restricción de acceso**: Usuario(Registrado) y Administrador
* **URI**: http://api.videostore.com/users/saldo
* **Método HTTP**: POST
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + id → identificador id del usuario.
  + valor→ cantidad a !!añadir!!. Se puede utilizar valores negativos para restar
* **Respuesta**: Información del usuario después de añadir
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.user+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Obtención lista películas compradas/alquiladas

* **Restricción de acceso**: Usuario/admin
* **URI**: http://api.videostore.com/user/{id}/movies
* **Método HTTP**: GET
* **Campos del formulario**
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Lista de las películas compradas/alquiladas por el usuario.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.movies\_user+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Recurso Película
2. Creación de Película

* **Restricción de acceso**: Administrador.
* **URI**: http://api.videostore.com/movie
* **Método HTTP**: POST
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
* (@FormParam("titulo") String titulo, @FormParam("genero") String genero,@FormParam("ano") int año, @FormParam("director") String director, @FormParam("descripcion") String descripcion, @FormParam("votos") int votos, @FormParam("numdesc") int numerodescargaspermitidas, @FormParam("tempmax") int tiempomaximovisualizacion, @FormParam("palquiler") int precioalquiler, @FormParam("pcompra") int preciocompra
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Información de la película creada.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.movie+json
* **Código de estado HTTP**: 201 Created
* **Errores**

400 - Bad Request → si no se envían todos los parámetros.

* + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Obtención de la información de la película

**Restricción de acceso**: Ninguno.

* **URI**: http://api.videostore.com/movie/{id}
* **Método HTTP**: GET
* **Respuesta**: Información de la película consultada.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.movie+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Añadir tu voto a una película

* **Restricción de acceso**: Usuario.
* **URI**: http://api.videostore.com/movie/vote/{id}
* **Método HTTP**: POST
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + ID Usuario→ Parámetro asociado a la película que la identifica.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Información de la película votada.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.movie+json
* **Código de estado HTTP**: 202 Accepted
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Búsqueda de películas

* **Restricción de acceso**: Ninguno.
* **URI**: http://api.videostore.com/movies/movie/search?tipo=titulo&valor=Avatar
* **Método HTTP**: GET
* **Tipo de medio de la petición**: QUERY\_PATH

Tipos (Con Valores, valores para la búsqueda):

- Tipo: Ano (año) Valor: “fecha pelicula”

- Tipo: titulo Valor: “nombre pelicula”

-Tipo: director Valor: “nombre director”

Tipo (Sin valotes, ORDER BY):

-Tipo: last (últimas entradas al videostore)

- Tipo: valoradas (top por puntación de votos)

-Tipo: destacadas (destacadas combinado por votos y últimas)

* **Respuesta**: Información de la listas de películas consultadas.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.movies\_collection+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Actualización de Película

* **Restricción de acceso**: Administrador.
* **URI**: http://api.videostore.com/movie/{id}
* **Método HTTP**: PUT
* **Tipo de medio de la petición**:application/vnd.dsa.videostore.movie+json
* **Campos del formulario**
* "buycost": 0,
* "description": "A paraplegic marine dispatched to the moon Pandora on a unique mission becomes torn between following his orders and protecting the world he feels is his home.",
* "director": "James Cameron",
* "genre": "SciFI",
* "id": "606ACF63B3A011E58635001F3BC3E069",
* "maxtimeshow": 3,
* "nummaxdownloads": 3,
* "rentcost": 12,
* "timeadded": 1451993709000,
* "title": "Avatar",
* "votes": 0,
* "year": 2012
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Información de la película creada.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.movie+json
* **Código de estado HTTP**: 201 Created
* **Errores**

400 - Bad Request → si no se envían todos los parámetros.

* + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.
* Eliminación de una película
* **Restricción de acceso**: Usuario(Registrado) y Administrador
* **URI**: http://api.videostore.com/movie/{id}
* **Método HTTP**: DELETE
* **Campos del formulario**
  + Token → Parámetro asociado a la cuenta que verifica la autentificación, Token de administrador en el caso del admin
* **Respuesta**: Vacia
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Recurso Alquiler
2. Alquiler de una película

* **Restricción de acceso**: Usuario.
* **URI**: http://api.videostore.com/alquiler/
* **Método HTTP**: POST
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Confirmación del alquiler de la película.
* **Código de estado HTTP**: 202 Accepted
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Actualización de una película

* **Restricción de acceso**: Admin.
* **URI**: http://api.videostore.com/alquiler/
* **Método HTTP**: PUT
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Horas restantes → Parámetro asociado a las horas restantes del alquiler
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Confirmación de la actualización
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.rent+json
* **Código de estado HTTP**: 202 Accepted
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Borrar una película

* **Restricción de acceso**: Admin.
* **URI**: http://api.videostore.com/alquiler/
* **Método HTTP**: DELETE
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Info alquiler de una película

* **Restricción de acceso**: Admin.
* **URI**: http://api.videostore.com/alquiler/{idusuario}/{idmovie}
* **Método HTTP**: GET
* **Tipo de medio de la petición**: PathParam
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Información del alquiler para el administrador
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.rent+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Recurso Compra

Compra de una película

* **Restricción de acceso**: Usuario.
* **URI**: http://api.videostore.com/comprar/
* **Método HTTP**: POST
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Confirmación del alquiler de la película.
* **Código de estado HTTP**: 202 Accepted
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Actualización de una película

* **Restricción de acceso**: Admin.
* **URI**: http://api.videostore.com/comprar/
* **Método HTTP**: PUT
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Descargas restantes → Parámetro asociado a las descargas restantes del la compra del usuario
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Confirmación de la actualización
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.rent+json
* **Código de estado HTTP**: 202 Accepted
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Borrar una película

* **Restricción de acceso**: Admin.
* **URI**: http://api.videostore.com/comprar/
* **Método HTTP**: DELETE
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Info compra de una película

* **Restricción de acceso**: Admin.
* **URI**: http://api.videostore.com/comprar/{idusuario}/{idmovie}
* **Método HTTP**: GET
* **Tipo de medio de la petición**: PathParam
* **Campos del formulario**
  + ID Película → Parámetro asociado a la película que la identifica.
  + ID Usuario→ Parámetro asociado al usuario que la identifica.
  + Token → Parámetro asociado a la cuenta que verifica la autentificación.
* **Respuesta**: Información del alquiler para el administrador
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.rent+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Recurso Login
2. Login

* **num);Restricción de acceso**: Ninguno.
* **URI**: http://api.videostore.com/login
* **Método HTTP**: POST
* **Tipo de medio de la petición**: application/x-www-form-urlencoded
* **Campos del formulario**
  + login id → Parámetro asociado a la cuenta que la identifica.
  + password → Parámetro asociado a la cuenta que la identifica.
* **Respuesta**: El token de acceso.
* **Tipo de medio de la respuesta**: application/vnd.dsa.videostore.user+json
* **Código de estado HTTP**: 200 OK
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.

1. Logout

* **Restricción de acceso**: Borrar el Token actual del usuario creado o logueado
* **URI**: http://api.videostore.com/logout
* **Método HTTP**: DELETE
* **Respuesta**: Vacía.
* **Código de estado HTTP**: 204 No Content
* **Errores**
  + 400 - Bad Request → si no se envían todos los parámetros.
  + 409 - Conflict → el identificador de *login* proporcionado por el usuario ya existe.