Aparmentshare

Diseño del servicio

http://eetac.upc.edu/ca/sites/all/themes/epsc/logo_eetac.png

Jordi López

Rubén Molina

Marcelus Zeron

Diseño de Servicios y Aplicaciones

Grupo: dsaqt1516g12

Contenido

[Definición 3](#_Toc440712272)

[Requisitos 4](#_Toc440712273)

[Recursos 7](#_Toc440712274)

[Usuario 7](#_Toc440712275)

[Piso 7](#_Toc440712276)

[Habitación 7](#_Toc440712277)

[Habitación Filtrada 8](#_Toc440712278)

[Campus 8](#_Toc440712279)

[Imagen de piso 8](#_Toc440712280)

[Imagen de habitación 8](#_Toc440712281)

[Correo Electrónico 9](#_Toc440712282)

[Mensaje 9](#_Toc440712283)

[Colección de Pisos 9](#_Toc440712284)

[Colección de Habitaciones 10](#_Toc440712285)

[Colección de Habitaciones Filtradas 10](#_Toc440712286)

[Colección de Campus 10](#_Toc440712287)

[Colección de Imágenes de piso 11](#_Toc440712288)

[Colección de Imágenes de habitación 11](#_Toc440712289)

[Colección de Mensajes 11](#_Toc440712290)

[Login 11](#_Toc440712291)

[Raíz 12](#_Toc440712292)

[Contrato Uniforme 12](#_Toc440712293)

[Recurso Usuario 12](#_Toc440712294)

[Creación de usuario 12](#_Toc440712295)

[Obtención del perfil del usuario 13](#_Toc440712296)

[Actualización de usuario 13](#_Toc440712297)

[Eliminación de un usuario 14](#_Toc440712298)

[Recurso Login 14](#_Toc440712299)

[Login 14](#_Toc440712300)

[Logout 15](#_Toc440712301)

[Recurso Piso 15](#_Toc440712302)

[Creación de piso 15](#_Toc440712303)

[Obtención de un piso 16](#_Toc440712304)

[Obtención de todos los pisos 16](#_Toc440712305)

[Actualización de un piso 17](#_Toc440712306)

[Eliminación de un piso 17](#_Toc440712307)

[Recurso Habitación 17](#_Toc440712308)

[Creación de habitación 17](#_Toc440712309)

[Obtención de una habitación 18](#_Toc440712310)

[Obtención de todas las habitaciones 18](#_Toc440712311)

[Actualización de una habitación 19](#_Toc440712312)

[Eliminación de una habitación 19](#_Toc440712313)

[Recurso Habitación Filtrada 20](#_Toc440712314)

[Creación de habitación Filtrada 20](#_Toc440712315)

[Obtención de una habitación 20](#_Toc440712316)

[Obtención de todas las habitaciones 21](#_Toc440712317)

[Recurso Campus 21](#_Toc440712318)

[Creación de un campus 21](#_Toc440712319)

[Obtención de un campus 22](#_Toc440712320)

[Obtención de todos los campus 22](#_Toc440712321)

[Actualización de un campus 22](#_Toc440712322)

[Eliminación de un campus 23](#_Toc440712323)

[Recurso Imágenes de piso 23](#_Toc440712324)

[Creación de una imagen de piso 23](#_Toc440712325)

[Obtención de todas las imágenes del piso 24](#_Toc440712326)

[Eliminación de una imagen de piso 24](#_Toc440712327)

[Recurso Imágenes de habitación 24](#_Toc440712328)

[Creación de una imagen de habitación 24](#_Toc440712329)

[Obtención de todas las imágenes de la habitación 25](#_Toc440712330)

[Eliminación de una imagen de habitación 25](#_Toc440712331)

[Recurso Imágenes de habitación filtrada 26](#_Toc440712332)

[Obtención de todas las imágenes de la habitación filtrada 26](#_Toc440712333)

[Recurso SendEmail 26](#_Toc440712334)

[Creación de un email 26](#_Toc440712335)

[Recurso Message 27](#_Toc440712336)

[Creación de un mensaje 27](#_Toc440712337)

[Obtención de un mensaje 27](#_Toc440712338)

[Obtención de mensajes 28](#_Toc440712339)

# Definición

El servicio está orientado a facilitar el contacto entre esos usuarios que desean alojarse cerca de los campus universitarios sin que les suponga un gran coste y usuarios que alquilan habitaciones en sus cercanías. Los usuarios registrados podrán poner en alquiler las habitaciones de sus pisos e interactuar con otros usuarios registrados que estén interesados en alguna de sus habitaciones. El servicio ayudará a interacturar entre usuarios registrados, en cambio un usuario no registrado sólo podrá obtener información.

# Requisitos

Requisitos a nivel funcional del servicio.

* Se definen tres tipos de usuario: los registrados*,* los no registrados y los administradores.
* Los **usuarios** no registrados podrán registrarse en el servicio. Los campos del registro son:
  + Un identificador de *login* que debe ser único. Por ejemplo, marceluz.
  + Una contraseña para acceder al servicio.
  + Un nombre completo. Por ejemplo Bob Esponja.
  + Una dirección de correo electrónico.
  + Teléfono de contacto.
* La autenticación de usuarios se realizará a través de un ***token*** de acceso (*access token*) que generará el servicio ante la presentación correcta de las credenciales de usuario (par identificador de login, contraseña).
* Un usuario registrado podrá crear ***pisos y habitaciones***. Un piso consta de:
  + El identificador de usuario que lo creó. Este identificador de usuario lo generará el servicio cuando el usuario se registre y será distinto al identificador de login.
  + Un campo *address* que dé título al *piso* y que tendrá una longitud máxima de 100 caracteres.
  + Un campo *description* que corresponde al contenido del *piso* y que tendrá una longitud máxima de 1000 caracteres.
  + Un campo con la marca temporal de la fecha de creación del *piso*.
  + Un campo con la marca temporal de la fecha de la última actualización del *piso*.
  + Un campo asociado al campus más cercano al piso.
  + Un campo que informa del número máximo de personas conviviendo juntas en el piso.
  + Un campo que informa de si se permite o no fumar en el piso.
  + Un campo que informa de si está permitido o no tener mascotas.
  + Un campo para informar si en el piso se permite sólo chicos, chicas o ambos.
  + Un campo para informar de los metros cuadrados del piso.
  + Un campo para informar si el piso está amueblado o no.
  + Un campo que informa del número de habitaciones que tiene el piso.
  + Un campo que informa del número de cuartos de baño disponibles en el piso.
  + Un campo para informar si el edificio donde se ubica el piso dispone de ascensor.
  + Un campo para informar si el piso dispone de acceso a Internet.
  + Un campo para informar de la fianza a dar al alquilar una de las habitaciones del piso.
  + Un campo para informar de la estancia mínima en meses que se debe respetar.
* Una **habitación** consta de:
  + El identificador de usuario que lo creó. Este identificador de usuario lo generará el servicio cuando el usuario se registre y será distinto al identificador de login.
  + El identificador del piso al que pertenece la habitación.
  + Un campo con la marca temporal de la fecha de creación de la habitación.
  + Un campo con la marca temporal de la fecha de la última actualización de la habitación.
  + Un campo para informar si la habitación está destinada a chico, chica o es indistinto.
  + Un campo para informar de los metros cuadrados de la habitación.
  + Un campo para informar si la habitación está amueblada o no.
  + Un campo para informar el estado actual de la habitación, si está o no ocupada actualmente.
  + Un campo que informa del precio mensual de la habitación.
* Un usuario registrado que haya creado un piso o habitación puede subir imágenes de la misma para que los usuarios interesados puedan verlas.
* Los pisos y habitaciones mostrarán su ubicación física a través de google maps para facilitar al interesado la distancia que tiene respecto al campus más cercano.
* Un usuario registrado creador puede actualiza todos los campos que definen el contenido de cualquier *piso o habitación* que haya creado. Excepto la fecha de creación, modificación y el usuario que lo creó.
* Un usuario registrado creador puede eliminar cualquier *piso* o habitación que haya creado.
* Cualquier usuario, registrado, puede obtener cualquier *piso* individual o colecciones paginables siempre y cuando lo haya creado él.
* Cualquier usuario, registrado o no, puede obtener cualquier *habitación* individual o colecciones activas paginables aunque no lo haya creado él.
* Cualquier usuario, registrado o no, puede filtrar los listados de habitaciones que le aparezcan para encontrar su habitación activa ideal.
* Un usuario administrador hará de moderador de la página. Estos son los encargados de eliminar pisos o habitaciones en caso de que lo crea conveniente (por ejemplo, un piso o habitación fraudulenta).
* Un usuario administrador es el único que puede crear, modificar y eliminar campus.
* Los usuarios registrados podrán comunicarse entre sí mediante mensajes internos o envío de un correo electrónico.

# Recursos

A continuación definiremos los recursos identificados para **aparmentshare**.

## Usuario

Representa a un usuario registrado en el sistema. De la definición de los requisitos sólo se deduce la operación de creación del usuario. Otras operaciones como modificación del perfil, eliminación, búsquedas, etc. quedan fuera de la implementación. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.user+json.

## Piso

Representa el contenido que crean los usuarios registrados. De los requisitos se deducen que sobre él se podrán realizar todas las operaciones CRUD: creación, obtención, actualización y eliminación. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.flat+json.

## Habitación

Representa el contenido que crean los usuarios registrados y dependen directamente de un piso. De los requisitos se deducen que sobre él se podrán realizar todas las operaciones CRUD: creación, obtención, actualización y eliminación. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.room+json.

## Habitación Filtrada

Representa el contenido de habitaciones activas que pueden ver todos los usuarios registrados y no registrados en función de sus criterios de búsqueda o disponibilidad. De los requisitos se deducen que sobre él se podrán realizar las operaciones CRUD: filtrar y obtener. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.room.filter+json.

## Campus

Representa el contenido que crea el Administrador para que los usuarios que creen pisos puedan asociarle el campus más cercano. De los requisitos se deducen que sobre él es que sólo el administrador podrá realizar todas las operaciones CRUD: creación, obtención, actualización y eliminación. El usuario registrado o no registrado podrá obtener el campus o campus que estén creados. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.campus+json.

## Imagen de piso

Representa el contenido que crean los usuarios registrados y depende directamente el piso al cual se asocia. De los requisitos se deducen que sobre él se podrán realizar las operaciones CRUD: creación, obtención, y eliminación. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.flat.img+json.

## Imagen de habitación

Representa el contenido que crean los usuarios registrados y depende directamente la habitación al cual se asocia. De los requisitos se deducen que sobre él se podrán realizar las operaciones CRUD: creación, obtención, y eliminación. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.room.img+json.

## Correo Electrónico

Representa el contenido generado por un usuario registrado al querer contactar con el usuario creador de una habitación. El creador recibirá un correo electrónico con el comentario del usuario interesado y además podrán mantener una conversación mediante correos electrónicos. El primer correo se quedará guardado como mensaje interno. De los requisitos se deducen que sobre él se podrán realizar la operación CRUD: creación. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.send.email+json.

## Mensaje

Representa el contenido generado por un usuario registrado a otro usuario registrado para mantener una conversación. De los requisitos se deducen que sobre él se podrán realizar la operación CRUD: creación. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.message+json.

## Colección de Pisos

Representa un conjunto de *pisos* que se devuelven como respuesta a peticiones de obtención cuyo resultado no es único. Así, por ejemplo, si se consultan todos los *pisos* creados por un usuario en concreto, se devolverían **todos** los *pisos* que ha creado el usuario. Como **todos** pueden ser muchos, en los requisitos se ha marcado que las colecciones deben ser paginables, es decir, que su contenido es un subconjunto de todos los *pisos* que hacen cierta la condición de discriminación, pero que contienen información suficiente como para poder continuar obteniendo más *pisos* que también cumplen la condición. Únicamente cada usuario registrado puede ver la colección de sus pisos. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.flat.collection+json.

## Colección de Habitaciones

Representa un conjunto de *habitaciones* que se devuelven como respuesta a peticiones de obtención cuyo resultado no es único. Así, por ejemplo, si se consultan todas las habitaciones creadas por un usuario en concreto, se devolverían **todas** las habitaciones que ha creado el usuario. Como **todas** pueden ser muchas, en los requisitos se ha marcado que las colecciones deben ser paginables, es decir, que su contenido es un subconjunto de todas las *habitaciones* que hacen cierta la condición de discriminación, pero que contienen información suficiente como para poder continuar obteniendo más *habitaciones* que también cumplen con las condiciones introducidas por el usuario interesado. Únicamente cada usuario registrado puede ver la colección de sus habitaciones. Este recurso se representará con el tipo de medio application/vnd. aparmentshare.room.collection+json.

## Colección de Habitaciones Filtradas

Representa un conjunto de *habitaciones* activas que se devuelven como respuesta a peticiones de obtención cuyo resultado no es único. Así, por ejemplo, si se consultan todas las habitaciones creadas en un determinado campus o con unos ciertos requisitos, se devolverían **todas** las habitaciones que coinciden con esas condiciones. Como **todas** pueden ser muchas, en los requisitos se ha marcado que las colecciones deben ser paginables, es decir, que su contenido es un subconjunto de todas las *habitaciones* que hacen cierta la condición de discriminación, pero que contienen información suficiente como para poder continuar obteniendo más *habitaciones* que también cumplen con las condiciones introducidas por el usuario interesado. Este recurso se representará con el tipo de medio application/vnd. aparmentshare.room.filter.collection+json.

## Colección de Campus

Representa un conjunto de *campus* que se devuelven como respuesta a peticiones de obtención cuyo resultado no es único. Así, por ejemplo, si se consultan todos los campus creados, se devolverían **todos** los campus que ha creado el administrador. Este recurso se representará con el tipo de medio application/vnd. aparmentshare.campus.collection+json.

## Colección de Imágenes de piso

Representa un conjunto de imágenes de *piso* que se devuelven como respuesta a peticiones de obtención cuyo resultado no es único. Así, por ejemplo, si se consultan todas las imágenes creadas asociadas a un piso, se devolverían **todas** las imágenes del *piso* que hay asociadas. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.flat.img.collection+json.

## Colección de Imágenes de habitación

Representa un conjunto de imágenes de *habitación* que se devuelven como respuesta a peticiones de obtención cuyo resultado no es único. Así, por ejemplo, si se consultan todas las imágenes creadas asociadas a una habitación, se devolverían **todas** las imágenes de la habitación que hay asociadas. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.room.img.collection+json.

## Colección de Mensajes

Representa un conjunto de mensajes intercambiados por usuarios registrados. Así, por ejemplo, si se consulta todos los mensajes, devolverá la fecha y hora del último mensaje, el número de mensajes totales y agrupado por remitente. Posteriormente se puede consultar todos los mensajes mantenidos con un determinador remitente. Este recurso se representará con el tipo de medio application/vnd.aparmentshare.message.collection+json.

## Login

Este recurso permite que los usuarios hagan *login* y *logout* en el servicio. El *login* permite a un usuario que presenta sus credenciales como el par identificador de usuario y contraseña obtener un *token* de acceso. La operación de *logout* elimina el *token* de acceso por lo que el usuario deberá volver a hacer*login* para obtener un nuevo. El recurso *token* de acceso se representa con el tipo de medio application/vnd.dsa.aparmentshare.auth-token+json.

## Raíz

Se define un recurso raíz a partir del cuál los consumidores del servicio (los clientes) puedan obtener enlaces que permitan *operar* sobre los recursos. Este recurso se representará con el tipo de medio application/vnd. aparmentshare.root+json.

# Contrato Uniforme

## Recurso Usuario

### Creación de usuario

**Restricción de acceso:** Ninguna.

**URI:** users.

**Método HTTP**: POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* loginid → Identificador único de login del usuario.
* password → Contraseña de acceso del usuario.
* email→ Dirección de correo electrónico del usuario.
* fullname→ Nombre completo del usuario.
* Phone 🡪 Teléfono de contacto del usuario.

**Respuesta:** Token de autorización asignado al usuario para realizar peticiones protegidas.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.auth-token+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* Location → el valor indica la URI del recurso usuario creado.

**Errores:**

* 400 - Bad Request → Si no se envían todos los parámetros.
* 409 - Conflict → El identificador de login proporcionado por el usuario ya existe.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un usuario por algún tipo de problema interno en el servidor.

El valor de la cabecera Location es igual a la URI del recurso usuario recién creado. Es habitual que cuando se cree un recurso aparte de indicar esa URI como uno de los enlaces del recurso devuelto se incluya, además, en esa cabecera HTTP. Se devuelve también una cabecera ETag para implementar peticiones HTTP condicionales sobre el recurso. De todos modos, sobre este recurso no se hace ninguna petición de lectura tal y como se ha especificado en la cabecera, por lo que podríamos obviarla.

### Obtención del perfil del usuario

**Restricción de acceso:** Ninguna.

**URI:** users/{id} donde {id} es el identificador del usuario del cual queremos obtener el perfil.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Recurso solicitado por el usuario.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.user+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ningún usuario con el identificador de usuario pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un usuario por algún tipo de problema interno en el servidor.

### Actualización de usuario

**Restricción de acceso:** Sólo el propio usuario puede modificar sus datos.

**URI:** users/{id} donde {id} es el identificador del usuario del cual queremos modificar el perfil.

**Método HTTP:** PUT

**Tipo de medio de la petición:** application/vnd.dsa.aparmentshare.user+json

**Entidad de la petición:** Recurso usuario. Se modificarán los campos de la dirección de correo electrónico, número de teléfono y/o el nombre completo.

**Respuesta:** Recurso de usuario modificado.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.user+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 400 - Bad Request → El parámetro identificador de usuario de la URL no coincide con el parámetro identificador de usuario de la entidad de la petición.
* 400 - Bad Request → Si la entidad está vacía o no está bien formada.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta modificar el perfil de otro usuario (No tiene autorización).
* 404 - Not Found → No hay ningún usuario con el identificador de usuario pasado en la ruta.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un usuario por algún tipo de problema interno en el servidor.

### Eliminación de un usuario

**Restricción de acceso:** Sólo el propio usuario puede eliminar su registro.

URI: users/{id} donde {id} es el identificador del usuario del cual queremos eliminar.

**Método HTTP:** DELETE

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Vacía.

**Código de estado HTTP:** 204 No Content

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta eliminar el perfil de otro usuario.
* 404 - Not Found → No hay ningún usuario con el identificador de usuario pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un usuario por algún tipo de problema interno en el servidor.

## Recurso Login

### Login

**Restricción de acceso:** Ninguna.

**URI:** login

**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.

**Respuesta:** El token de acceso.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.auth-token+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 400 - Bad Request → Si no se envían todos los parámetros.
* 400 - Bad Request → Si no existe el identificador de login.
* 400 - Bad Request → Si la contraseña no coincide con la del usuario.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un usuario por algún tipo de problema interno en el servidor.

### Logout

**Restricción de acceso:** Un usuario registrado que actualmente tenga un token de acceso concedido.

**URI:** login

**Método HTTP:** DELETE

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Vacía.

**Código de estado HTTP**: 204 No Content

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un usuario por algún tipo de problema interno en el servidor.

## Recurso Piso

### Creación de piso

**Restricción de acceso:** Registrado

**URI:** flat

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* location → Dirección física del piso.
* description → Breve descripción del piso.
* numpartner → Número máximo de personas en el piso.
* smoker → Si está permitido fumar o no en el piso.
* pets → Si está permitido tener mascota en el piso.
* girlorboy → Si el piso acepta sólo chicos, chicas o ambos.
* sqm → Metros cuadrados habitables del piso.
* furnished → Si el piso está amueblado o no.
* numrooms → El número de habitaciones de las que dispone el piso.
* elevator → Si el piso dispone de ascensor.
* plantnum → La altura en la que se encuentra el piso.
* internet → Si el piso dispone de acceso a Internet.
* numbathrooms →El número de baños del piso
* fianza → El dinero de fianza que se debe dar al coger una de las habitaciones del piso.
* estancia → El número mínimo de meses que se debe estar al coger una de las habitaciones del piso.

**Respuesta:** El piso creado

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.flat+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso piso creado.

**Errores:**

* 400 - Bad Request → Si no se envían los parámetros obligatorios.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → El campus seleccionado no existe.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un piso por algún tipo de problema interno en el servidor.

### Obtención de un piso

**Restricción de acceso:** Sólo el usuario que creó el piso.

**URI:** flat/{id} donde {id} es el identificador del piso que queremos obtener.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** El piso pedido.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.flat+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta obtener el piso de otro usuario (No tiene autorización).
* 500 - Internal Server Error → Error genérico cuando no se puede obtener un piso por algún tipo de problema interno en el servidor.

### Obtención de todos los pisos

**Restricción de acceso:** Sólo el usuario que creó el piso.

**URI:** flat

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todos los pisos.

**Tipo de medio de la respuesta:** application/vnd.dsa.beeter.aparmentshare.flat.collection+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un piso por algún tipo de problema interno en el servidor.

### Actualización de un piso

Restricción de acceso: Sólo el usuario que creó el piso puede modificarlo.

URI: flat/{id} donde {id} es el identificador del piso que queremos modificar.

**Método HTTP:** PUT

**Tipo de medio de la petición:** application/vnd.dsa.aparmentshare.flat+json

**Entidad de la petición:** Recurso piso. Se modificarán los campos de descripción, número de compañeros, fumador, mascotas, chicos/chicas, metros cuadrados, amueblado, localización, número de baños, ascensor, planta, internet, fianza, estancia mínima y/o descripción.

**Respuesta:** Recurso piso modificado.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.flat+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 400 - Bad Request → El parámetro identificador de piso de la URL no coincide con el parámetro identificador de piso de la entidad de la petición.
* 400 - Bad Request → Si la entidad está vacía o no está bien formada.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta modificar un piso creado por otro usuario.
* 404 - Not Found → No hay ningún piso con el identificador pasado en la ruta.
* 500 - Internal Server Error → Error genérico cuando no se puede actualizar un piso por algún tipo de problema interno en el servidor.

### Eliminación de un piso

**Restricción de acceso:** Sólo el propio usuario que creó un piso puede eliminarlo.

**URI:** flat/{id} donde {id} es el identificador del piso del cual queremos eliminar.

**Método HTTP:** DELETE

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Vacía.

**Código de estado HTTP:** 204 No Content

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta eliminar el piso de otro usuario.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede crear un piso por algún tipo de problema interno en el servidor.

## Recurso Habitación

### Creación de habitación

**Restricción de acceso:** Registrado

**URI:** flat/{id}/room donde el {id} es el identificador del piso que contiene la habitación.

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* description → Breve descripción de la habitación.
* girlorboy → Si el piso acepta sólo chicos, chicas o ambos.
* sqm → Metros cuadrados de la habitación del piso.
* furnished → Si la habitación está amueblada o no.
* status → Si la habitación está libre o alquilada.
* price → El precio mensual de la habitación.

**Respuesta:** La habitación creada

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.room+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso habitación creada.

**Errores:**

* 400 - Bad Request → Si no se envían los parámetros obligatorios.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 403 - Forbidden → Un usuario intenta añadir una habitación en el piso de otro usuario.
* 500 - Internal Server Error → Error genérico cuando no se puede crear la habitación por algún tipo de problema interno en el servidor.

### Obtención de una habitación

**Restricción de acceso:** Sólo el usuario que creó la habitación.

**URI:** flat/{id}/room/{id} donde el primer {id} es el identificador del piso y el segundo {id} es el identificador de la habitación la cual queremos obtener. (Para usuarios registrados)

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** La habitación solicitada.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.room+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta obtener una habitación de otro usuario.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener la habitación por algún tipo de problema interno en el servidor.

### Obtención de todas las habitaciones

**Restricción de acceso:** Sólo el usuario que creó la habitación.

**URI:** flat/{id}/room donde el primer {id} es el identificador del piso el cual queremos obtener todas sus habitaciones.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todas las habitaciones.

**Tipo de medio de la respuesta:** application/vnd.dsa.beeter.aparmentshare.room.collection+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener el conjunto de habitaciones por algún tipo de problema interno en el servidor.

### Actualización de una habitación

**Restricción de acceso:** Sólo el usuario que creó la habitación puede modificarlo.

**URI:** flat/{id}/room/{id} donde el primer {id} es el identificador del piso y el segundo {id} es el identificador de la habitación la cual queremos actualizar.

**Método HTTP:** PUT

**Tipo de medio de la petición:** application/vnd.dsa.aparmentshare.room+json

**Entidad de la petición:** Recurso habitación. Se modificarán los campos de descripción, chicas/chicos, metros cuadrados, amueblado, estado, precio y/o descripción.

**Respuesta:** Recurso habitación modificado.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.room+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 400 - Bad Request → El parámetro identificador de la habitación de la URL no coincide con el parámetro identificador de habitación de la entidad de la petición.
* 400 - Bad Request → Si la entidad está vacía o no está bien formada.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta modificar una habitación creada por otro usuario.
* 404 - Not Found → No hay ninguna habitación con el identificador pasado en la ruta.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede actualizar la habitación por algún tipo de problema interno en el servidor.

### Eliminación de una habitación

**Restricción de acceso:** Sólo el propio usuario que creó una habitación puede eliminarla.

**URI:** flat/{id}/room/{id} donde el primer {id} es el identificador del piso y el segundo {id} es el identificador de la habitación la cual queremos eliminar.

**Método HTTP:** DELETE

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Vacía.

**Código de estado HTTP:** 204 No Content

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta eliminar la habitación de otro usuario.
* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede eliminar la habitación por algún tipo de problema interno en el servidor.

## Recurso Habitación Filtrada

### Creación de habitación Filtrada

**Restricción de acceso:** Ninguna.

**URI:** rooms

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* smoker → Si en el piso se puede fumar, no u es indiferente.
* girlorboy → Si la habitación busca chica, chico o es indiferente.
* pets → Si el piso acepta o no animales de compañía o es indiferente.
* numpartner→ El número de compañeros que hay en el piso
* minprice → El precio mínimo mensual por la habitación.
* maxprice→ El precio máximo mensual por la habitación.
* campusname→ El ID del campus vinculado al piso de la habitación.

**Respuesta:** Las habitaciones activas que cumplan con las condiciones añadidos, ningún campo del formulario es obligatorio.

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.room.filter.collection+json

**Código de estado HTTP:** 200 OK

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso de las habitaciones listadas.

**Errores:**

* 500 - Internal Server Error → Error genérico cuando no se puede crear la habitación por algún tipo de problema interno en el servidor.

### Obtención de una habitación

**Restricción de acceso:** Ninguna.

**URI:** rooms/{id} donde el primer {id} es el identificador del de la habitación la cual queremos obtener.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** La habitación solicitada.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.room.filter+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener la habitación por algún tipo de problema interno en el servidor.

### Obtención de todas las habitaciones

**Restricción de acceso:** Ninguna.

**URI:** rooms.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todas las habitaciones activas.

**Tipo de medio de la respuesta:** application/vnd.dsa.beeter.aparmentshare.room.filter.collection+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener el conjunto de habitaciones por algún tipo de problema interno en el servidor.

## Recurso Campus

### Creación de un campus

**Restricción de acceso:** Administrador

**URI:** campus

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* campusname → El nombre del campus.
* address → Dirección física del campus.
* longitud → Coordenadas geográficas basada en la línea del ecuador del planeta.
* latitud → Coordenadas geográficas basada en la línea del Meridiano de Greenwich del planeta.

**Respuesta:** El campus creado.

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.campus+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso campus creado.

**Errores:**

* 400 - Bad Request → Si no se envían todos los parámetros.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 500 - Internal Server Error → Error genérico cuando no se puede crear el campus por algún tipo de problema interno en el servidor.

### Obtención de un campus

**Restricción de acceso:** Ninguna.

**URI:** campus/{id} donde el {id} es el identificador del campus el cual queremos obtener.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** El campus solicitado.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.campus+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ninguna campus con el identificador de campus pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener el campus por algún tipo de problema interno en el servidor.

### Obtención de todos los campus

**Restricción de acceso:** Ninguna.

**URI:** campus

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todos los campus.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.campus.collection+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 500 - Internal Server Error → Error genérico cuando no se puede obtener el conjunto de campus por algún tipo de problema interno en el servidor.

### Actualización de un campus

**Restricción de acceso:** Administrador.

**URI:** campus/{id} donde el {id} es el identificador del campus el cual queremos actualizar.

**Método HTTP:** PUT

**Tipo de medio de la petición:** application/vnd.dsa.aparmentshare.campus+json

**Entidad de la petición:** Recurso campus. Se modificarán los campos del nombre del campus, localización, longitud y latitud.

**Respuesta:** Recurso campus modificado.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.campus+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 400 - Bad Request → El parámetro identificador del campus de la URL no coincide con el parámetro identificador del campus de la entidad de la petición.
* 400 - Bad Request → Si la entidad está vacía o no está bien formada.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No hay ningún campus con el identificador pasado en la ruta.
* 500 - Internal Server Error → Error genérico cuando no se puede actualizar el campus por algún tipo de problema interno en el servidor.

### Eliminación de un campus

**Restricción de acceso:** Administrador.

**URI:** campus/{id} donde el {id} es el identificador del campus el cual queremos eliminar.

**Método HTTP:** DELETE

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Vacía.

**Código de estado HTTP:** 204 No Content

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No hay ningún campus con el identificador de campus pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede eliminar el campus por algún tipo de problema interno en el servidor.

## Recurso Imágenes de piso

### Creación de una imagen de piso

**Restricción de acceso:** Sólo el usuario que creó el piso.

**URI:** flat/{id}/img donde el id es el identificador del piso.

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* location → Ubicación de la imagen a subir.

**Respuesta:** Imagen creada.

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.flat.img+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso imagen creada.

**Errores:**

* 400 - Bad Request → Si no se envían todos los parámetros.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta subir una imagen en un piso que no es suyo.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede subir la imagen por algún tipo de problema interno en el servidor.

### Obtención de todas las imágenes del piso

**Restricción de acceso:** Sólo el usuario que creó el piso.

**URI:** flat/{id}/img donde el {id} es el identificador del piso la cual queremos obtener todas las imágenes.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todas las imágenes.

**Tipo de medio de la respuesta:** application/vnd.dsa.beeter.aparmentshare.flat.img.collection+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 403 - Forbidden → Un usuario intenta visualizar imágenes en un piso que no es suyo.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener el conjunto de imágenes del piso por algún tipo de problema interno en el servidor.

### Eliminación de una imagen de piso

**Restricción de acceso:** Sólo el usuario que creó el piso.

**URI:** flat/{id}/img/{id} donde el {id} es el identificador del piso y el segundo {id} identifica la imagen del piso la cual queremos obtener.

**Método HTTP:** DELETE

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Vacía.

**Código de estado HTTP:** 204 No Content

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 403 - Forbidden → Un usuario intenta eliminar una imagen de piso de otro usuario.
* 404 - Not Found → No hay ninguna imagen con el identificador de imagen pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede eliminar la imagen por algún tipo de problema interno en el servidor.

## Recurso Imágenes de habitación

### Creación de una imagen de habitación

**Restricción de acceso:** Sólo el usuario que creó la habitación.

**URI:** flat/{id}/room/{id}/img

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* location → Ubicación de la imagen a subir.

**Respuesta:** Imagen creada.

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.room.img+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso imagen creada.

**Errores:**

* 400 - Bad Request → Si no se envían todos los parámetros.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 403 - Forbidden → Un usuario intenta crear una imagen de habitación de otro usuario.
* 500 - Internal Server Error → Error genérico cuando no se puede subir la imagen por algún tipo de problema interno en el servidor.

### Obtención de todas las imágenes de la habitación

**Restricción de acceso:** Sólo el usuario que creó la habitación.

**URI:** flat/{id}/room/{id}/img donde el {id} es el identificador del piso y el segundo {id} identifica la habitación la cual queremos obtener todas las imágenes.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todas las imágenes.

**Tipo de medio de la respuesta:** application/vnd.dsa.beeter.aparmentshare.room.img.collection+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 403 - Forbidden → Un usuario intenta visualizar imágenes de la habitación de otro usuario.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener el conjunto de imágenes de la habitación por algún tipo de problema interno en el servidor.

### Eliminación de una imagen de habitación

**Restricción de acceso:** Sólo el usuario que creó la habitación.

**URI:** flat/{id}/room/{id}/img/{id} donde el {id} es el identificador del piso y el segundo {id} identifica la habitación y el tercero la imagen de la habitación la cual queremos obtener.

**Método HTTP:** DELETE

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Vacía.

**Código de estado HTTP:** 204 No Content

**Errores:**

* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No hay ningún piso con el identificador de piso pasado.
* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 403 - Forbidden → Un usuario intenta eliminar una imagen de habitación de otro usuario.
* 500 - Internal Server Error → Error genérico cuando no se puede eliminar la imagen por algún tipo de problema interno en el servidor.

## Recurso Imágenes de habitación filtrada

### Obtención de todas las imágenes de la habitación filtrada

**Restricción de acceso:** Ninguna.

**URI:** rooms/{id}/img donde el {id} es el identificador de la habitación la cual queremos obtener todas las imágenes.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todas las imágenes de la habitación.

**Tipo de medio de la respuesta:** application/vnd.dsa.beeter.aparmentshare.room.img.collection+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener el conjunto de imágenes de la habitación por algún tipo de problema interno en el servidor.

## Recurso SendEmail

### Creación de un email

**Restricción de acceso:** Usuario registrado.

**URI:** rooms/{roomid}/email donde el {roomid} es el ID de la room.

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* bodytext → El texto que se quiere enviar al creador de la habitación.

**Respuesta:** El campus creado.

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.send.email+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso sendemail creado.

**Errores:**

* 400 - Bad Request → Si no se envían todos los parámetros.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No hay ninguna habitación con el identificador de habitación pasado.
* 404 - Not Found → No existe el usuario que quiere enviar el email.
* 404 - Not Found → No existe el usuario creador de la habitación.
* 500 - Internal Server Error → Error genérico cuando no se puede crear el campus por algún tipo de problema interno en el servidor.

## Recurso Message

### Creación de un mensaje

**Restricción de acceso:** Usuario registrado.

**URI:** message

**Método HTTP:** POST

**Tipo de medio de la petición:** application/x-www-form-urlencoded

**Campos del formulario:**

* text → El texto que se quiere enviar al usuario destinatario del mensaje.
* Loginid → El loginid del usuario que debe recibir el mensaje.

**Respuesta:** El campus creado.

**Tipo de medio de la respuesta:** application/vnd.dsa.apartmentshare.message+json

**Código de estado HTTP:** 201 Created

**Cabeceras adicionales:**

* location → El valor indica la URI del recurso mensaje creado.

**Errores:**

* 400 - Bad Request → Si no se envían todos los parámetros.
* 401 - Unauthorized → Si la petición no presenta el token de acceso.
* 404 - Not Found → No existe el loginid introducido.
* 500 - Internal Server Error → Error genérico cuando no se puede crear el campus por algún tipo de problema interno en el servidor.

### Obtención de un mensaje

**Restricción de acceso:** Usuario registrado.

**URI:** message/{id} donde el {id} es el identificador del usuario que ha recibido nuestros mensajes.

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** El campus solicitado.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.messages+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 404 - Not Found → No hay ninguna usuario con el identificador de usuario pasado.
* 500 - Internal Server Error → Error genérico cuando no se puede obtener el campus por algún tipo de problema interno en el servidor.

### Obtención de mensajes

**Restricción de acceso:** Usuario registrado.

**URI:** message

**Método HTTP:** GET

**Tipo de medio de la petición:** Ninguno.

**Respuesta:** Todos los campus.

**Tipo de medio de la respuesta:** application/vnd.dsa.aparmentshare.message+json

**Código de estado HTTP:** 200 OK

**Errores:**

* 500 - Internal Server Error → Error genérico cuando no se puede obtener el conjunto de campus por algún tipo de problema interno en el servidor.