

Rutas para API de TASKS

<http://mibaldi.com/swagger/dist/index.html>

El siguiente API trabaja con usuarios que disponen de una serie de campos, como son su username y password, el teléfono, el nombre, la edad y también tiene asociada una lista de emails y lista de tareas.

Cada usuario se creara mediante el API, y para poder modificar o trabajar con las tareas tendrá que realizar un login.

A la hora de realizar el login el usuario meterá su username y su password y se le devolverá un token que tendrá que añadir en la cabecera de las siguientes consultas.

El usuario podrá ver su información, modificarla y borrarse a si mismo junto con las tareas que esta llevando a cabo en ese momento. También podrá visualizar la lista de los usuarios registrados con una información básica de estos, para realizar esta acción no se requiere estar registrado en el API.

Una vez registrado el usuario puede crear, borrar y editar sus propias tareas.

Las tareas disponen de un campo llamado titulo que sera el identificador de la tarea, una descripción, un campo de fecha final y una lista de tags, que serán palabras clave que compatiran las tareas como puede ser el tag "Compras" o el tag "VACACIONES", los tags se añadirán a tareas creadas.

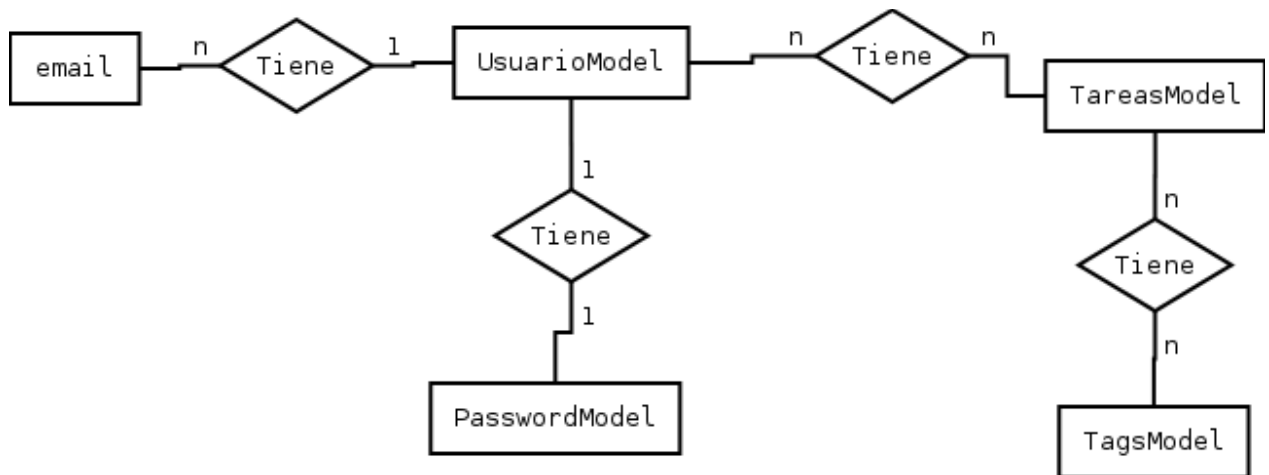
Un usuario perteneciente a una tarea tiene el poder para añadir a otros usuarios a esa tarea.

El usuario dispondrá de la posibilidad de visualizar una tarea concreta o listar las tareas de su cuenta. Tiene dos opciones de listar las tareas, una por usuarios y otra por tags.

La lista de tareas por usuario mostrara las tareas del usuario registrado en ese momento y se podrán filtrar por las tareas que se comparten con otros usuarios.

Si el usuario quiere filtrar las tareas por tags, podrá listar las tareas que cumplan unos tags definidos y se le mostraran todas las tareas que cumplan algún tag de los escritos.

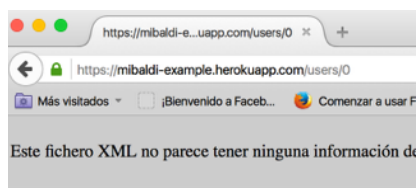
Diagrama UML



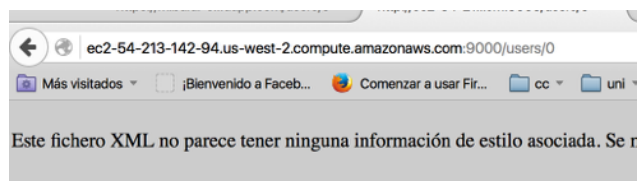
Despliegue

El despliegue se ha realizado tanto en heroku como en amazon.

A continuación se muestran capturas del API funcionando en las dos plataformas



```
- <usuarios>
  <count>1</count>
  - <usuario id="12">
    <username>mikel</username>
    <nombre>mibaldi</nombre>
    <edad>0</edad>
    <telefono></telefono>
  - <tareas>
    <tarea>tarea1</tarea>
  </tareas>
  <emails></emails>
</usuario>
</usuarios>
```



```
- <usuarios>
  <count>1</count>
  - <usuario id="1">
    <username>mikel2</username>
    <nombre>mibaldi</nombre>
    <edad>0</edad>
    <telefono></telefono>
  </tareas>
  <emails></emails>
</usuario>
</usuarios>
```

Documentación

Para la realización de la documentación del API se ha usado el modulo de Swagger-play, que mediante anotaciones produce un fichero JSON con la información de las rutas e información importante sobre cada función.

Para que se muestre de una manera mas vistosa para el usuario se ha usado el proyecto de swagger-ui mediante el cual lee el fichero JSON que devuelve el API en una ruta definida y lo muestra al usuario mediante colores y desplegables.

La ruta donde se ha colgado el proyecto donde se muestra la información es un servidor distinto al que se ha colgado el API.

El servidor donde esta colgada la web de visualización es:

- <http://mibaldi.com/swagger/dist/index.html>

El servidor desde donde coge la información a mostrar es:

- <https://mibaldi-example.herokuapp.com>

Swagger Usuario-Tareas MIMO

La direccion del API es: <https://mibaldi-example.herokuapp.com>

tarea Show/Hide | List Operations | Expand Operations

Method	Path	Description
POST	/tarea	crear tarea
PUT	/tareaTags/{texto}/{tag}	añade tag a tarea
PUT	/tareaUsuario/{texto}/{usuario}	añade usuario a tarea
GET	/tareas	get lista tareas usuarios
DELETE	/tareas/{texto}	borrar tarea
GET	/tareas/{texto}	get tarea
PUT	/tareas/{texto}	modificar tarea
GET	/tareasTag	get lista tareas tag

user Show/Hide | List Operations | Expand Operations

Method	Path	Description
POST	/login	logearse
GET	/logout	desconectarse

El API acepta tanto el valor “en” como es “es” en las cabeceras de Accept-Language.

Se han realizado los test tanto de los controladores como de los modelos principales, Usuarios y Tareas.

Usuario

- Login : POST

- <https://mibaldi-example.herokuapp.com/login>
- Formato del body:

```
{
  "username": "mibaldi",
  "pass": { "contrasenia": "12345678" }
}
```

- Content-Type=application/json
- La contraseña una vez se ha creado el usuario se encripta y no puede volverse a modificar. En la base de datos se guarda el valor de la contraseña encriptada para compararla en la sección de login.
- Response:

Response Messages

HTTP Status Code	Reason
200	TOKEN
400	Invalid format
404	Usuario o password incorrecto

- Logout : GET

- <https://mibaldi-example.herokuapp.com/logout>
- Response:

Response Messages

HTTP Status Code	Reason
200	Desconectado

- Crear usuario: POST

- <https://mibaldi-example.herokuapp.com/user>
- Formato del body:

```
{
  "username": "mibaldi",
  "pass": { "contrasenia": "12345678" },
  "nombre": "mikel balduciel",
  "edad": 24,
  "telefono": "666777888",
  "emails": [ { "email": "mibaldi2@hotmail.com" } ]
}
```

- Content-Type=application/json
- Response :

Response Messages

HTTP Status Code	Reason
200	Creacion correcta
409	Usuario ya existente

- Ver usuario: GET

- <https://mibaldi-example.herokuapp.com/user>
- Response:

Response Messages

HTTP Status Code	Reason
200	Usuario
400	Invalid format
404	Usuario no logeado

- Actualizar usuario: PUT

- <https://mibaldi-example.herokuapp.com/user>
- Formato del body:

```
{
  "username": "mibaldi",
  "nombre": "Mikel",
  "telefono": "666777888",
  "edad": 24,
  "pass": {
    "contrasenia": "12345678"
  }
}
```

- Content-Type=application/json
- Response:

Response Messages

HTTP Status Code	Reason
200	Modificacion realizada
304	Sin modificaciones
400	Invalid format
404	Usuario no logeado

- Borrar usuario: DELETE

- <https://mibaldi-example.herokuapp.com/user>
- Response:

Response Messages

HTTP Status Code	Reason
200	Borrado realizado
404	Usuario no logeado

- Lista de usuarios: GET

- `https://mibaldi-example.herokuapp.com/users/<pag>?nombre=<nombre>&edad=<edad>`
- Los campos nombre y edad de la query son campos opcionales y filtra los usuarios de la lista por esas restricciones. La paginación empieza en 0 y se le puede agregar el campo size en la query para decidir la cantidad de usuarios que se mostraran por cada pagina.
- Response:

Response Messages

HTTP Status Code	Reason
200	Lista Usuarios
400	Invalid format

Tarea

- Ver lista tareas : GET

- <https://mibaldi-example.herokuapp.com/tareas?users=<username1,username2,username3>>
- En los parámetro de la query se pueden poner los usuarios con los que comparte tareas y la respuesta devolverá las tareas en las cuales están tanto el usuario logeado como los usuarios de la lista.
- Response:

Response Messages	
HTTP Status Code	Reason
200	Creacion correcta
400	Invalid tarea
404	Usuario no logeado

- Ver tarea : GET

- <https://mibaldi-example.herokuapp.com/tareas/<tituloTarea>>
- Response:

Response Messages	
HTTP Status Code	Reason
200	Tarea
400	Invalid format
401	No existe la tarea para el usuario logeado
404	Usuario no logeado

- Ver lista tareas por Tag : GET

- <https://mibaldi-example.herokuapp.com/tareasTag?tags=<tag1,tag2,tag3>>
- En los parámetro de la query se pueden poner los tags de las tareas y la respuesta devolverá las tareas en las cuales están los tags puestos en la query
- Response:

Response Messages	
HTTP Status Code	Reason
200	Lista Tareas por tag
400	Invalid format
404	Usuario no logeado

- Crear tarea : POST

- <https://mibaldi-example.herokuapp.com/tarea>
- Formato del body:

```
{  
  "titulo": "tarea1",  
  "descripcion": "comprar el pan",  
  "fechafin": "2015-12-25 12:00:00"  
}
```

- Content-Type=application/json
- Response:

Response Messages	
HTTP Status Code	Reason
200	Creacion correcta
400	Invalid tarea
404	Usuario no logeado

- Modificar tarea : PUT

- <https://mibaldi-example.herokuapp.com/tareas/<tarea>>
- Formato del body:

```
{  
  "titulo": "tarea1",  
  "descripcion": "comprar el pan",  
  "fechafin": "2015-12-25 12:00:00"  
}
```

- Content-Type=application/json
- Response:

Response Messages	
HTTP Status Code	Reason
200	Modificacion realizada
304	Sin modificaciones
400	Invalid tarea
401	No existe la tarea para el usuario logeado
404	Usuario no logeado

- Añadir usuario a tarea : PUT

- <https://mibaldi-example.herokuapp.com/tareaUsuario/<tarea>/<usuario>>
- Response:

Response Messages	
HTTP Status Code	Reason
200	Usuario añadido
401	No existe la tarea para el usuario logeado
404	Usuario no logeado

- Añadir tag a tarea : PUT

- <https://mibaldi-example.herokuapp.com/tareaTags/<texto>/<tag>>
- Response:

Response Messages	
HTTP Status Code	Reason
200	Tag añadido
401	No existe la tarea para el usuario logeado
404	Usuario no logeado

- Delete tarea : DELETE

- <https://mibaldi-example.herokuapp.com/tareas/<texto>>
- Response:

Response Messages	
HTTP Status Code	Reason
200	Usuario añadido
401	No existe la tarea para el usuario logeado
404	Usuario no logeado