AURORA – Manual Programador

# 

# **Control de Documento**

| Información General del Documento |  |
| --- | --- |
| Versión: | 1.0 |
| Estado de Versión: | Actualización |
| Fecha Versión: | 13/07/2021 |

| Log de Revisión | | | |
| --- | --- | --- | --- |
| Versión | Fecha Revisión | Motivo de Cambio | Autor |
| 1 | 13/07/2021 | Creación | Mariano Pagani |
| 1 | 18/06/2022 | Actualizacion | Mariano Pagani |
|  |  |  |  |

| Distribución/ Historia de aprobación | | | |
| --- | --- | --- | --- |
| Versión revisada | Aprobador | Firma | Fecha Aprobación |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introducción

## Sobre la aplicación

**Aurora** es una aplicación para que el turista que visita Formosa pueda acceder de forma fácil a los lugares turísticos que ofrece la provincia.

Está dividido en una app mobile, que permite al turista loguearse, completar con sus gustos e intereses para luego ofrecerle de un mapa en donde le muestra los lugares sugeridos de acuerdo a sus preferencias que puede visitar.

Por otro lado existe una App web la cual sirve para administrar las categorías y lugares que el cliente puede buscar. También en esta app mobile se van a poder visualizar reportes que agrupa nicho de usuarios con ciertos gustos.

## Tecnologías

### Aplicación Web

La aplicación web es un SPA realizada con Javascript con la biblioteca de Reactjs.

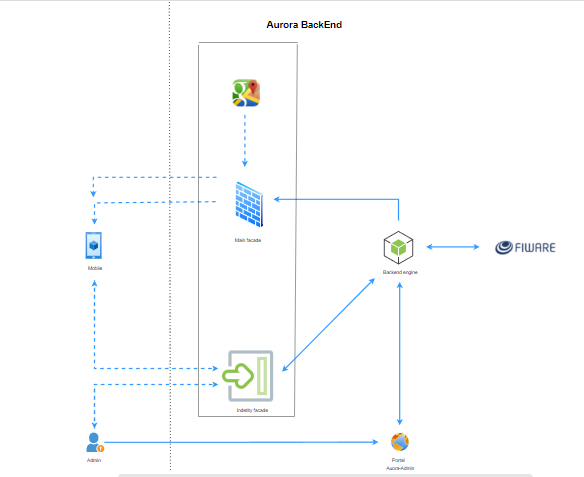
### Aplicación Mobile

La app Mobile fue realizada con Kotlin.

### BackEnd

El backEnd fue realizado con Node Js utilizando el framework Express. La construcción de los modelos de Categorías y Puntos de interés se hizo a través de los smart models proporcionados por fiware. También se utilizó Keyrock para la autenticación y autorización de usuarios. Para persistir los puntos de interés se utilizó el servicio Orion.

## Arquitectura del Proyecto



[*https://drive.google.com/file/d/1x8fRB4jbZHl1bQJxU6rgTrezlHqThL6L/view?usp=sharing*](https://drive.google.com/file/d/1x8fRB4jbZHl1bQJxU6rgTrezlHqThL6L/view?usp=sharing)

Principalmente tenemos dos flujos: El flujo de usuario administrador, el cual accede al portal para administrar toda la información que consume el usuario de la app mobile

El otro flujo que tenemos es el de usuario, el cual mediante la app mobile accede a la aplicación.

Ambos flujos se comunican con el servidor de NodeJs el cual tiene principalmente dos endPoints importantes realizados con el patrón de desarrollo Facade:

1. Facade de Identidad

Permite a todos los usuarios del ecosistemas loguearse para acceder al porta/App web

1. Main Facade

Es el punto de acceso de la app Mobile y de la app web para consumir todos los modelos ofrecidos.

# Entidades

### Categorías

#### Descripción

Son las categorías a las cuales puede pertenecer un punto de interés.

#### Propiedades

**Index**: Entero que va desde el 1 hasta el 484

**Value**: Contiene dentro la propiedad “labels”. Cada Labels tiene 10 propiedades más las cuales representan los acrónimos de los lenguajes que tienen disponible, cada uno con su descripción

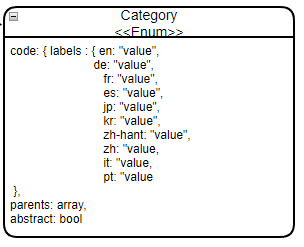
**Parents**: Es un array de tipo entero que contiene los índices relacionados al objeto actual

**Abstract:** Es un booleano, por defecto siempre se encuentra en verdadero.

#### 

#### 

#### Modelo



### Puntos de Interés

#### Descripción

Son los puntos de interés que un turista puede ver en el mapa

#### Propiedades

**Id**: es el identificador único del objeto. Autogenerado

**Type:** Siempre va ser de tipo PointOfInterest. Solo lectura

**DataProvider:**  Especifica la url con información acerca del proveedor de información. Opcional

**DateModified:** Es la fecha de modificación. Solo lectura.

**DateCreated:** La fecha de creación.Solo lectura.

**Name**: El nombre del punto de interés. Obligatorio.

**Description**: La descripción del punto de interés. Opcional.

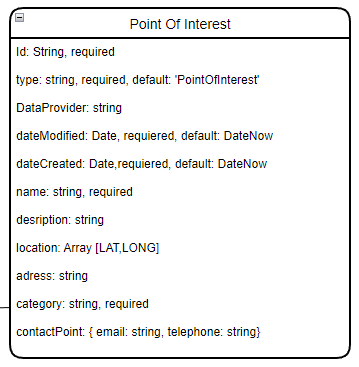
**Location:** Son las coordenadas del punto de interés, es obligatorio si no se manda la dirección.

**Address:** Es la dirección del punto de interés. Es obligatorio si no se manda la localización.

**Category:** Es la categoría del punto de interés. Obligatorio

**ContactPoint:** Es información sobre contacto sobre el punto de interés.Opcional.

#### Modelo



### 

### **Sugerencias**

#### Descripción

Son los comentarios enviados por el usuario de la app mobile.

#### Propiedades

**Id**: es el identificador único del objeto.

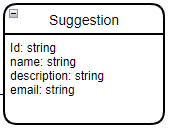
**Name**: Es el nombre de la persona que envía la sugerencia

**Descripción:** Son los comentarios enviados por el usuario.

**Email**: Es el email del usuario que envía la sugerencia

#### 

#### Modelo



### **Usuarios**

#### Descripción

Son los usuarios que pueden acceder al sistema

#### Propiedades

**Id**: es el identificador único del objeto.

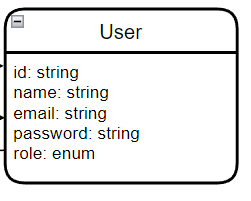
**Name**: Es el nombre del usuario.

**Email**: Es el email del usuario

**Password**: Es el password del usuario

#### 

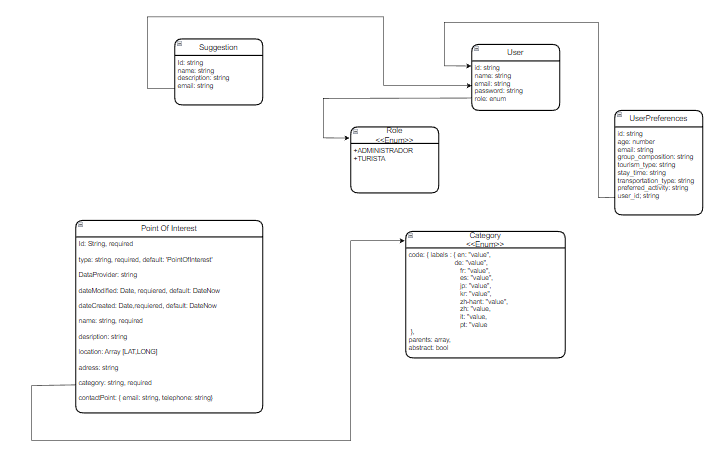
#### Modelo



# Diagrama Entidad-Relación

En la siguiente página se encuentra un esquema que representa el diagrama y las relaciones del sistema Aurora.

*ver imagen en la página siguiente*



[*https://drive.google.com/file/d/1LowoNyU8bKsBPLrnwWR4AGl-nfqdg-Km/view?usp=sharing*](https://drive.google.com/file/d/1LowoNyU8bKsBPLrnwWR4AGl-nfqdg-Km/view?usp=sharing)

# 

# Diagrama de Clases

# 

# 

[*https://drive.google.com/file/d/16lLThz7QS-\_HArMmSfSMsbci\_wpwWvY4/view?usp=sharing*](https://drive.google.com/file/d/16lLThz7QS-_HArMmSfSMsbci_wpwWvY4/view?usp=sharing)

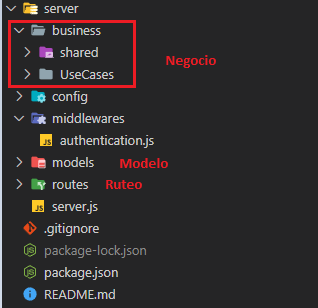
# Servidor

## 

## Arquitectura

El servidor fue realizado con una arquitectura en capas. El mismo cuenta con la capa de negocios, la capa de modelo y la capa de ruteo.

## 



La capa de ruteo se encarga de recibir y redirigir los request entrantes al caso de uso correspondiente.

La capa de negocios contiene toda la lógica del negocio y es la que resuelve los request.

La capa de modelo es en donde se encuentran los objetos que representan las entidades

## 

## Requisitos

1- Para trabajar en forma local se necesita tener levantado docker con los servicios de MongoDb, MySql, Orion y KeyRock.

2- Configurar la url de Fiware keyrock y Orion. Para más información por favor vea la sección configuración de entorno.

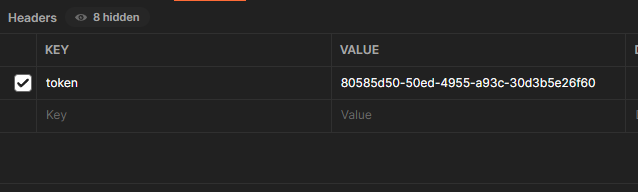
## API

### Autenticación y Autorización.

A excepción de los métodos para registrarse o loguearse, el resto de los endpoints son privados.

Se deberá enviar en los headers una key token, con el token recibido al loguearse.

Ejemplo:



Existe dos tipos de roles que un usuario puede tener:  
  
**ADMIN\_ROLE**: Rol asignado a las personas que trabajan con el portal web. Un usuario con este rol puede realizar las acciones de un CRUD (create, read, updated and delete) sobre todas las entidades.

**USER\_ROLE**: Este rol pertenece a los usuarios de la app mobile. Pueden realizar todas las acciones relacionadas a listar información. También son los únicos habilitados a enviar sugerencias

### Manejo de errores

Las salidas de éxito/fallo se gestionarán a través de códigos de estado HTTP.

La solicitud exitosa obtendrá un HTTP 200 y la respuesta resultante.

Si no se procesa la solicitud, se indicará mediante un código de estado de rango de HTTP 400 junto con un objeto llamado err que contiene el mensaje del error.

##### 

### Métodos soportados

###### **Category**

Endpoint para obtener categorías

Métodos disponibles:

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Category | /category/{Id} | Obtener una categoría por id | **GET** | USER\_ROLE, ADMIN\_ROLE |
| Categories | /categories | Obtener todas las categorías | **GET** | USER\_ROLE, ADMIN\_ROLE |
| CategoryValues | /categoryValues | Devuelve todos los Id de las categorías | **GET** | USER\_ROLE, ADMIN\_ROLE |

###### **Point of Interest (poi)**

Endpoint para administrar los puntos de interés.

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Point Of Interest | /pointsOfInterest | Crear un poi | **POST** | ADMIN\_ROLE |
| Get single | /pointOfInterest/:id | obtiene un poi | **GET** | USER\_ROLE, ADMIN\_ROLE |
| Get many | /pointsOfInterest | obtiene todos los poi | **GET** | USER\_ROLE, ADMIN\_ROLE |
| Delete | /pointOfInterest/:id | elimina un poi | **DEL** | ADMIN\_ROLE |
| Update | /pointOfInterest/:id | actualiza un poi | **PUT** | ADMIN\_ROLE |

###### **Suggestion**

Endpoint para crear u obtener sugerencias.

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Create single | /suggestion | Crea una sugerencia | **POST** | USER\_ROLE |
| Suggestions | /suggestions | Obtiene todas las sugerencias | **GET** | ADMIN\_ROLE |

###### **Search**

Endpoint para buscar por nombre o por categoría un punto de interés.

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Search POI | /search | Busca un punto de interés | **GET** | USER\_ROLE, ADMIN\_ROLE |

###### 

###### **Users**

Endpoint para administrar los usuarios del sistema. Solo se pueden crear usuarios con el Rol de ADMIN\_ROLE. No se pueden editar usuarios con USER\_ROLE, pero si se pueden listar.

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Get many | /user | Obtener todos los usuarios | **GET** | ADMIN\_ROLE |
| Get single | /user/{id} | obtener un usuario | **GET** | ADMIN\_ROLE |
| Create | /user | crear un usuario | **POST** | ADMIN\_ROLE |
| Update | /user{id} | actualizar un usuario | **PUT** | ADMIN\_ROLE |
| Delete | /user{id} | eliminar un usuario | **DEL** | ADMIN\_ROLE |

###### **User Preferences**

Permite al usuario administrar las preferencias de su perfil.

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Get | /user-preferences | Obtener todas las preferencias | **GET** | ADMIN\_ROLE |
| Get single | /user/:{id} | obtener una preferencia | **GET** | ADMIN\_ROLE USER\_ROLE |
| Create | /user-preferences | crear una preferencia | **POST** | USER\_ROLE |
| Update | /user-preferences/:{id} | actualizar una preferencia | **PUT** | USER\_ROLE |
| Delete | /user-preferences/:{id} | eliminar una preferencia | **DEL** | ADMIN\_ROLE |

###### 

###### **Reports**

Permite al usuario obtener un reporte.

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Get | /report | Obtener un reporte | **GET** | ADMIN\_ROLE |

###### 

###### **Identity**

Permite al usuario registrarse o loguearse.

| **Nombre** | **Path** | **Descripción** | **HTTP Verb** | **ROLE** |
| --- | --- | --- | --- | --- |
| Login | /login | Permite loguearse en el servidor y obtener un token | **POST** | - |
| Register | /register | Permite registrar un usuario con rol de USER\_ROLE | **POST** | - |

### Estructura de los mensajes

###### **Category**

**GET CategoryValues**

**URL: http://localhost:3000/categoryValues**

**EXAMPLE**

curl --location --request GET 'http://localhost:3000/categoryValues'

**GET Categories**

**URL http://localhost:3000/categories**

**PARAMS**

from: int (optional, default 0)

count: int (optional, default 5)

**EXAMPLE**

curl --location --request GET 'http://localhost:3000/categories?from=1&limit=5'

**GET Category**

**URL http://localhost:3000/category/83**

curl --location --request GET 'http://localhost:3000/category/83'

###### **Point of Interest**

**POST pointsOfInterest**

**URL: http://localhost:3000/pointsOfInterest**

**HEADERS**

key: token

value : {{token}}

**BODY  
  
{**

**"category": string , *Required***

**"dataProvider": string ,**

**"name": string , *Required***

**"description": string ,**

**"location": Array, *Required \****

**"address": string , *Required \****

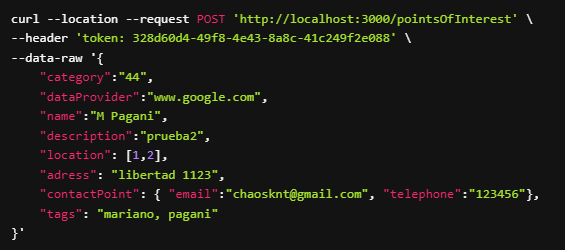
**"contactPoint": { "email": string , "telephone": string },**

**"tags": string**

**}**

***\**** *Se debe enviar al menos uno de los dos, el que no se mande es opcional.*

**EXAMPLE**



**GET Get single**

**URL** http://localhost:3000/pointOfInterest/:{id}

EXAMPLE

curl --location --request GET 'http://localhost:3000/pointOfInterest/928b7b98-df4a-48a6-b27b-89e7a444bdca' \

--data-raw ''

**GET Get many**

**URL** http://localhost:3000/pointsOfInterests?from=3&limit=2

**PARAMS**

from: int (optional, default 0)

limit: int (optional, default 5)

**Example Request**

curl --location --request GET 'http://localhost:3000/pointsOfInterests?from=3&limit=2' \

--data-raw ''

**DEL Delete**

**URL** http://localhost:3000/pointOfInterest/:{id}

**HEADERS**

key: token

Value: {{token}}

**EXAMPLE  
  
curl --location --request DELETE 'http://localhost:3000/pointOfInterest/12f3a3af-67a5-4d3b-be7a-61cd2322a368' \**

**--data-raw ''**

**PUT Update**

**URL** http://localhost:3000/pointOfInterest/928b7:{id}

**HEADERS**

key: token

Value: {{token}**}**

**EXAMPLE**  
BODY

****

###### **Suggestion**

**POST Create single**

**URL** http://localhost:3000/suggestion

{

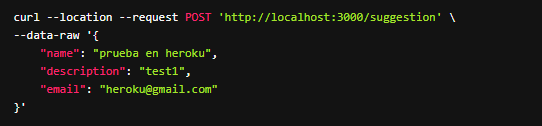
"name": "string", ***Required***

"description": "string", ***Required***

"email": "string" ***Required***

}

Example



**GET Suggestions**

**URL** http://localhost:3000/suggestions

**PARAMS**

from: int (optional, default 0)

limit: int (optional, default 5)

Example



###### **Users**

**GET Get many**

**URL** http://localhost:3000/user

**HEADERS**

key: token

Value: {{token}}

**EXAMPLE**

curl --location --request GET 'http://localhost:3000/user' \

**GET Get single**

**URL http://localhost:3000/user/:{id}**

**HEADERS**

key: token

Value: {{token}**}**

**EXAMPLE**

curl --location --request GET 'http://localhost:3000/user/b45864d6-badf-4da6-8268-7ed69fa60bed' \

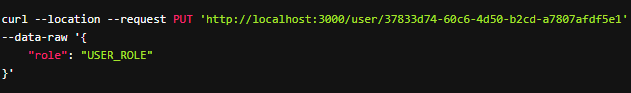
**PUT Update**

**URL http://localhost:3000/user/:{id}**

**{**

**"role": "enum"**

**}**

**Example  
  
**

**DEL Delete**

**URL http://localhost:3000/user/:{id}**

curl --location --request DELETE 'http://localhost:3000/user/c8b53617-c7de-4a55-bdb2-13232903a2cc' \

**POST Create**

**URL http://localhost:3000/user/**

**HEADERS**

key: token

Value: {{token}}

**Body:**

**{**

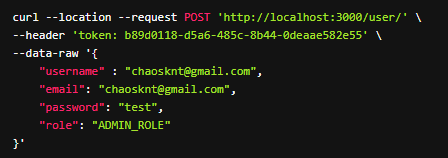
**"username" : "string",**

**"email": "string",**

**"password": "string",**

**"role": "enum"**

**}**

****

###### **User Preferences**

**GET Get**

**URL : http://localhost:3000/user-preferences**

**HEADERS**

key: token

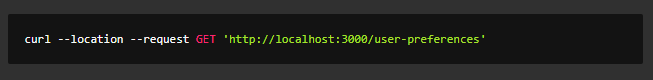
Value: {{token}}

**PARAMS**

from: string (optional, default 1) \*

limit: string (optional, default 5) \*

**Example**

****

###### 

**GET GetById**

**URL : http://localhost:3000/user-preferences/:{id}**

**HEADERS**

key: token

Value: {{token}}

****

**POST create**

**URL : http://localhost:3000/user-preferences**

**HEADERS**

key: token

Value: {{token}}

**{**

**"user\_id": "Guid",**

**"procedence": "string",**

**"age": string,**

**"email": "string",**

**"group\_composition": "ENUM",**

**"tourism\_type": "ENUM",**

**"stay\_time": "ENUM",**

**"transportation\_type": "ENUM",**

**"preferred\_activity": "ENUM"**

**}**

***En la sección Anexos se encuentra un aparatado con todos los valores posibles.***

**Example**

****

**PUT update**

**URL http://localhost:3000/user-preferences/:{id}**

**HEADERS**

key: token

Value: {{token}}

**{**

**"user\_id": "Guid",**

**"procedence": "string",**

**"age": string,**

**"email": "string",**

**"group\_composition": "ENUM",**

**"tourism\_type": "ENUM",**

**"stay\_time": "ENUM",**

**"transportation\_type": "ENUM",**

**"preferred\_activity": "ENUM"**

**}**

**Example**

****

**DEL Delete**

**URL http://localhost:3000/user-preferences/:{id}**

**HEADERS**

key: token

Value: {{token}}

****

###### 

###### **Reports**

**GET Get**

**HEADERS**

key: token

Value: {{token}}

**PARAMS**

from: Date obligatorio

to: Date obligatorio

**Example**

****

###### 

###### **Search**

**GET searchPoI**

**URL http://localhost:3000/search**

**PARAMS**

name: string obligatorio

lan: string (optional, default en) \*

\*Lan es el lenguaje en el cual esta especificada la busqueda

curl --location --request GET 'http://localhost:3000/search?name=museo&lan=en

###### **Identity**

**POST Login**

**URL http://localhost:3000/login**

**BODY**

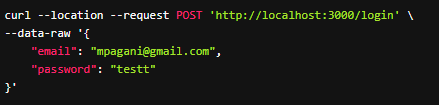
**{**

**"email": "string",**

**"password": "string"**

**}**

**Example**

****

**POST Register**

**URL http://localhost:3000/register**

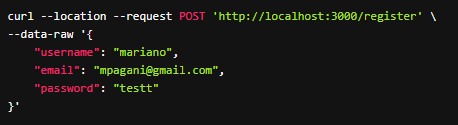
**{**

**"username": "string",**

**"email": "string",**

**"password": "string"**

**}**

****

# Requisitos del Sistema

## Requerimientos no funcionales

* Fiware
* App orientada teniendo en cuenta los tiempos del turista.
* OpenMaps (de ser posible)

## Requerimientos funcionales App Mobile

Que el usuario Turista

* Pueda loguear en la app mobile
* Pueda completar un wizard con sus preferencias
* Pueda visualizar en el mapa los lugares sugeridos
* Disponga de un buscador de lugares turísticos

## Requerimientos funcionales App web

Que el usuario Administrador

* Pueda loguearse.
* Disponga de una pantalla para la administración de usuarios.
* Disponga de una pantalla para la administración de categorías de lugares turísticos.
* Disponga de una pantalla para la administración de atracciones.
* Disponga de una pantalla para visualizar reportes

# Casos de prueba

## Servidor

*Por cuestiones de tiempo solo se realizaron los caso de uso más importantes, queda pendiente revisar el resto de casos de uso*

| **Caso de Uso** | **Prueba** | **Resultado esperado** | **Resultado obtenido** |
| --- | --- | --- | --- |
| Login | Loguearse con Usuario correcto, contraseña incorrecta | No puede loguear | No se loguea |
| Login | Loguearse con Usuario correcto, contraseña correcta | puede loguear | se loguea |
| Registrarse | Con usuario que no existe | se registra | se registra |
| Registrarse | Con usuario que existe | no se registra | no se registra |
| Crear punto de interés | Con datos correctos | Se crea el punto de interés | Se crea el punto de interés |
| Crear punto de interés | Con datos incorrectos | no se crea el punto de interés | no se crea el punto de interés |
| Actualizar punto de interés | Con datos correctos | Se crea el punto de interés | Se crea el punto de interés |
| Actualizar punto de interés | Con datos incorrectos | no se crea el punto de interés | no se crea el punto de interés |
| Eliminar punto de interés | Con id que existe | Se elimina el punto de interés | Se elimina el punto de interés |
| Eliminar punto de interés | Con id que no existe | no se elimina el punto de interés | no se elimina el punto de interés |
| Obtener todos los puntos de interés | Obtener todos los puntos de interés | Se listan todos los puntos de interés | Se listan todos los puntos de interés |
| Obtener punto de interés por id | Obtener punto de interés por id que existe | Se obtiene un punto de interés | Se obtiene un punto de interés |
| Obtener punto de interés por id | Obtener punto de interés por id que no existe | no se obtiene un punto de interés | no se obtiene un punto de interés |

# Configuración entorno

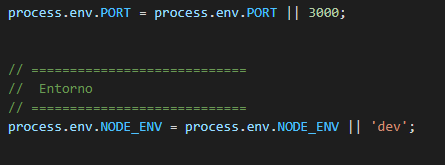
## 

## Servidor

Dentro de la carpeta del servidor se encuentra una carpeta config la cual tiene dos archivos: config y fiwareConfig.

### config

* process.env.PORT: es el puerto en el cual se va levantar el servidor, por default es 3000.
* process.env.NODE\_ENV: Es el entorno en el cual se está trabajando, por default es dev



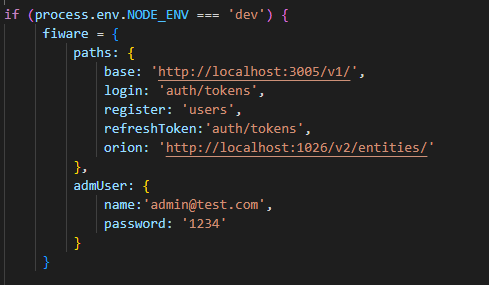
#### FiwareConfig

Contiene toda la configuración requerida para utilizar fiware. Se debe configurar:

* Paths: son todos los url que se necesitan para utilizar la aplicación, los mismos son:

1. base: url de fiware
2. login: url de autenticación
3. register: url de registro
4. refreshToken: url para actualizar el token
5. orion: url en donde se encuentra levantado orion.

* AdminUser : es el usuario administrador el cual es el que obtendrá el token de administrador para realizar las peticiones a fiware.



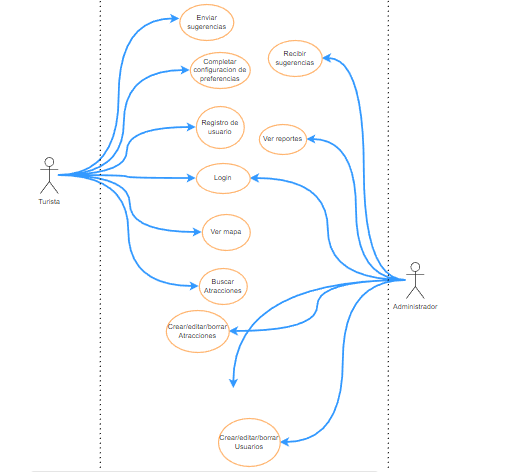
# 

# Anexos

## User Preferences Values

1. Composición del Grupo
2. Solo
3. pareja
4. amigos
5. familia
6. Tipo de turismo
7. Vacaciones
8. Negocios
9. Tiempo de permanencia en ciudad:
10. medio día
11. un día
12. fin de semana
13. otro
14. Medio de transporte en la ciudad
15. auto
16. bus,
17. caminando
18. Actividad preferida
19. Aire libre
20. Naturaleza
21. Cultura
22. Historia
23. Náutica
24. Recreación

## Casos de uso



[*https://drive.google.com/file/d/1UgVUgLoOwScQVFgPPcb-nf410KgJ11KQ/view?usp=sharing*](https://drive.google.com/file/d/1UgVUgLoOwScQVFgPPcb-nf410KgJ11KQ/view?usp=sharing)

Registro en la App Mobile

Yo como usuario turista quiero poder registrarme en app mobile para poder entrar y buscar atracciones

Login en la App Mobile

Yo como usuario turista quiero poder loguear en la app mobile para buscar atracciones.

Completar configuración de preferencias en la App Mobile

Yo como usuario turista quiero poder completar mis preferencias para que al loguear me muestre atracciones basadas en esa configuración.

Envío de sugerencias en la App Mobile (?)

Yo como usuario turista quiero poder enviar sugerencias para que sea incluidas en la app mobile.

Ver Mapa en la App Mobile

Yo como usuario turista quiero ver en un mapa con marcadores las atracciones ofrecidas.

Buscar Atracciones en la App Mobile

Yo como usuario turista quiero poder buscar atracciones.

Login en la App Web

Yo como usuario administrador quiero loguear en la app web para poder realizar mantenimiento.

Recibir sugerencias en la App Web (?)

Yo como usuario administrador quiero recibir sugerencias para poder mejorar las atracciones.

Ver Reportes en la App Web

Yo como usuario administrador quiero recibir reportes de los usuarios turistas para poder tomar mejores decisiones.

Alta-Baja-Modificación de Usuarios en la App Web

Yo como usuario administrador quiero poder dar de alta, modificar y eliminar otros usuarios administradores.

Alta-Baja-Modificación de Categorías en la App Web

Yo como usuario administrador quiero poder dar de alta, modificar y eliminar categorías de las atracciones.

Alta-Baja-Modificación de Atracciones en la App Web

Yo como usuario administrador quiero poder dar de alta, modificar y eliminar Atracciones.

## Credenciales y Urls del proyecto

### Repositorio

<https://github.com/aurora-tourism>

### Google

Email: [aurora.tourism.project@gmail.com](mailto:aurora.tourism.project@gmail.com)  
Password: Xj6hAz.K45l

### MongoDbAtlas

Email: aurora.tourism.project@gmail.com

Password: Xj6hAz.K45l

### Cluster credenciales

aurorabackend

rKPldZhTh4ix40B5

### Mongodb Url

'mongodb+srv://aurorabackend:GUW14yq9KX98xZYU@cluster0.mspev.mongodb.net/test'

### HEROKU

Email: aurora.tourism.project@gmail.com

clave: Xj6hAz.K45l

url: https://radiant-shore-31658.herokuapp.com/

https://git.heroku.com/radiant-shore-31658.git

### Postman documentation

<https://documenter.getpostman.com/view/13862293/UzBmLSTM>