

---

# DOCUMENTACION API

---

FORMAT: 1A

HOST: <http://emilio1991.pythonanywhere.com/api>

## # API Inventario

Inventario es una API que permite a los clientes poder gestionar los inventarios de cualquier tipo de negocio.

## #Sesiones

### ## Gestion de Sesiones

#### ### POST [/login]

Esta accion nos sirve para poder acceder al sistema y a los registros que ya tengamos asociados a este usuario o a crear nuevos registros

+ Response 200 (application/json)

+ Atributos (object)

+ usuario: Emilio (string)

+ contraseña: "123" (string)

+ Cuerpo

```
{  
  "message": "OK"  
}
```

#### ###GET [/logout]

Esta accion nos sirve para poder salir del sistema

+ Response 200 (application/json)

+ Cuerpo

```
{  
  "message": "OK"  
}
```

## #Usuarios

### ## Gestion de Usuarios

#### ### GET [/usuarios/{id\_usuario}]

Esta accion solo devuelve el Usuario con el "id\_usuario" que se especifica.

#### ### GET [/usuarios]

Esta accion devuelve todos los Usuarios que esten en la Base de Datos.

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ usuario: Emilio (string)

+ contraseña: "pbkdf2:sha256:50000\$JZu8m8AT..." (string, encriptada)

+ nombre: Luis Garcia (string)

+ descripcion: Usuario frecuente (string)

+ administrador: 1 (tinyint)

+ Cuerpo

```
{
  "administrador": 123,
  "contrasena":
"pbkdf2:sha256:50000$o6ZrPZnO$oc036810b17a74c3f349002coe1a794e4acdb73b91257e2
44da62a4f1d49b148",
  "descripcion": "123",
  "id": 1,
  "nombre": "123",
  "usuario": "123"
}
```

#### ### POST [/usuarios]

Esta accion se encarga de crear un nuevo registro en la tabla de "usuarios".

+ Response 200 (application/json)

+ Atributos (object)

- + id: 1 (int, requerido)
- + usuario: Emilio (string)
- + contraseña: "pbkdf2:sha256:50000\$JZu8m8AT..." (string, encriptada)
- + nombre: Luis Garcia (string)
- + descripcion: Usuario frecuente (string)
- + administrador: 1 (tinyint)

+ Cuerpo

```
{
  "administrador": 123,
  "contrasena":
"pbkdf2:sha256:50000$o6ZrPZnO$oc036810b17a74c3f349002coe1a794e4acdb73b91257e2
44da62a4f1d49b148",
  "descripcion": "123",
  "id": 1,
  "nombre": "123",
  "usuario": "123"
}
```

### ### PUT [/usuarios/{usuario\_id}]

Esta accion se encarga de modificar un registro en la tabla de "usuarios", según el "usuario\_id" que se le manda como parametro.

+ Response 200 (application/json)

+ Atributos (object)

- + id: 1 (int, requerido)
- + usuario: Emilio (string)
- + contraseña: "pbkdf2:sha256:50000\$JZu8m8AT..." (string, encriptada)
- + nombre: Luis Garcia (string)
- + descripcion: Usuario frecuente (string)
- + administrador: 1 (tinyint)

+ Cuerpo

```
{
  "administrador": 1,
```

```

        "contrasena":
"pbkdf2:sha256:50000$o6ZrPZnO$oc036810b17a74c3f349002coe1a794e4acdb73b91257e2
44da62a4f1d49b148",
        "descripcion": "Usuario Frecuente",
        "id": 1,
        "nombre": "Luis Garcia",
        "usuario": "Emilio"
    }

```

## #Inventarios

### ## Gestion de Inventarios

#### ### GET [/inventarios/{id\_inventario}]

Esta accion solo devuelve el Inventario con el "id\_inventario" que se especifica.

#### ### GET [/inventarios]

Esta accion devuelve todos los Inventarios que esten en la Base de Datos, según el usuario logeado.

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Inventario 1 (string)

+ usuario: Datos del Usuario al que pertenece el inventario

+ Cuerpo

```

{
    "id": 1,
    "nombre": "Inventario 1",
    "usuario": {
        "administrador": 1,
        "contrasena":
"pbkdf2:sha256:50000$o6ZrPZnO$oc036810b17a74c3f349002coe1a794e4acdb73b
91257e244da62a4f1d49b148",
        "descripcion": "Usuario Frecuente",
        "id": 1,
        "nombre": "Luis Garcia",
        "usuario": "Emilio"
    }
}

```

### ### POST [/inventarios]

Esta accion se encarga de crear un nuevo registro en la tabla de "inventario".

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Inventario 1 (string)

+ usuario: Datos del Usuario al que pertenece el inventario

+ Cuerpo

```
{
  "id": 1,
  "nombre": "Inventario 1",
  "usuario": {
    "administrador": 1,
    "contrasena":
      "pbkdf2:sha256:50000$o6ZrPZnO$oc036810b17a74c3f349002coe1a794e4acdb73b
      91257e244da62a4f1d49b148",
    "descripcion": "Usuario Frecuente",
    "id": 1,
    "nombre": "Luis Garcia",
    "usuario": "Emilio"
  }
}
```

### ### PUT [/inventarios/{inventario\_id}]

Esta accion se encarga de modificar un registro en la tabla de "inventario", según el "inventario\_id" que se le manda como parametro.

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Inventario 1 (string)

+ usuario: Datos del Usuario al que pertenece el inventario

+ Cuerpo

```

{
  "id": 1,
  "nombre": "Inventario 1",
  "usuario": {
    "administrador": 1,
    "contrasena":
    "pbkdf2:sha256:50000$o6ZrPZnO$oc036810b17a74c3f349002coe1a794e4acdb73b
    91257e244da62a4f1d49b148",
    "descripcion": "Usuario Frecuente",
    "id": 1,
    "nombre": "Luis Garcia",
    "usuario": "Emilio"
  }
}

```

## #Categorias

### ## Gestion de Categoriass

#### ### GET [/categorias/{id\_categoria}]

Esta accion solo devuelve el registro de la tabla "Categoria" con el "id\_categoria" que se especifica y que este asociado al usuario logueado.

#### #### GET [/categorias]

Esta accion devuelve todas las Categorias que esten en la Base de Datos, según el usuario logueado.

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Inventario 1 (string)

+ descripcion: Prueba (string)

+ inventario: Datos del Inventario al que pertenece el inventario

+ Cuerpo

```

{
  "id": 1,
  "nombre": "Categoria 1",
  "descripcion": "Prueba"
  "inventario": {
    "id": 1,

```

```

        "nombre": "Inventario 1" ,
        "usuario": 1
    }
}

```

### ### POST [/categorias]

Esta accion se encarga de crear un nuevo registro en la tabla de "categoria".

+ Response 200 (application/json)

+ Atributos (object)

```

+ id: 1 (int, requerido)
+ nombre: Inventario 1 (string)
+ descripcion: Prueba (string)
+ inventario: Datos del Inventario al que pertenece el inventario

```

+ Cuerpo

```

{
    "id": 1,
    "nombre": "Categoria 1" ,
    "descripcion": "Prueba"
    "inventario": {
        "id": 1,
        "nombre": "Inventario 1" ,
        "usuario": 1
    }
}

```

### ### PUT [/categorias/{categoria\_id}]

Esta accion se encarga de modificar un registro en la tabla de "Categoria", según el "categoria\_id" que se le manda como parametro.

+ Response 200 (application/json)

+ Atributos (object)

```

+ id: 1 (int, requerido)
+ nombre: Inventario 1 (string)

```

```

+ descripcion: Prueba (string)

+ inventario: Datos del Inventario al que pertenece el inventario

+ Cuerpo
{
    "id": 1,
    "nombre": "Categoria 1" ,
    "descripcion": "Prueba"
    "inventario": {
        "id": 1,
        "nombre": "Inventario 1" ,
        "usuario": 1
    }
}

```

## **#Proveedores**

### **## Gestion de Proveedores**

#### **### GET [/proveedores/{id\_proveedor}]**

Esta accion solo devuelve el registro de la tabla "Proveedor" con el "id\_proveedor" que se especifica y que este asociado al usuario logueado.

#### **### GET [/proveedores]**

Esta accion devuelve todos los Proveedores que esten en la Base de Datos, según el usuario logueado.

+ Response 200 (application/json)

```

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Proveedor 1 (string)

+ descripcion: Prueba (string)

+ Cuerpo
{
    "id": 1,
    "nombre": "Proveedor 1" ,
    "descripcion": "Prueba"
}

```



### ### POST [/proveedores]

Esta accion se encarga de crear un nuevo registro en la tabla de "proveedor".

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Proveedor 1 (string)

+ descripcion: Prueba (string)

+ Cuerpo

```
{
  "id": 1,
  "nombre": "Proveedor 1",
  "descripcion": "Prueba"
}
```

### ### PUT [/proveedor/{proveedor\_id}]

Esta accion se encarga de modificar un registro en la tabla de "proveedor", según el "proveedor\_id" que se le manda como parametro.

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Proveedor 1 (string)

+ descripcion: Prueba (string)

+ Cuerpo

```
{
  "id": 1,
  "nombre": "Proveedor 1",
  "descripcion": "Prueba"
}
```

## #Productos

### ## Gestion de Productos

#### ### GET [/productos/{id\_producto}]

Esta accion solo devuelve el registro de la tabla "Producto" con el "id\_producto" que se especifica y que este asociado al usuario logueado.

#### ### GET [/productos]

Esta accion devuelve todos los Productos que esten en la Base de Datos, según el usuario logueado.

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Producto 1 (string)

+ descripcion: Prueba (string)

+ categoria: 1 (int), Categoria al que esta asociado el registro

+ proveedor: 1 (int), Proveedor al que esta asociado el registro

+ Cuerpo

```
{
  "id": 1,
  "nombre": "Categoria 1",
  "descripcion": "Prueba",
  "categoria": {
    "id": 1,
    "nombre": "Categoria 1",
    "descripcion": "Prueba"
  },
  "inventario": 1
},
{
  "proveedor": {
    "id": 1,
    "nombre": "Proveedor 1",
    "descripcion": "Prueba"
  }
}
```

#### ### POST [/productos]

Esta accion se encarga de crear un nuevo registro en la tabla de "producto".

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Producto 1 (string)

+ descripcion: Prueba (string)

+ categoria: 1 (int), Categoria al que esta asociado el registro

+ proveedor: 1 (int), Proveedor al que esta asociado el registro

+ Cuerpo

```
{
  "id": 1,
  "nombre": "Categoria 1",
  "descripcion": "Prueba",
  "categoria": {
    "id": 1,
    "nombre": "Categoria 1",
    "descripcion": "Prueba",
    "inventario": 1
  },
  "proveedor": {
    "id": 1,
    "nombre": "Proveedor 1",
    "descripcion": "Prueba"
  }
}
```

### ### PUT [/productos/{producto\_id}]

Esta accion se encarga de modificar un registro en la tabla de "producto", según el "producto\_id" que se le mande como parametro.

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ nombre: Producto 1 (string)

- + descripcion: Prueba (string)
- + categoria: 1 (int), Categoria al que esta asociado el registro
- + proveedor: 1 (int), Proveedor al que esta asociado el registro

+ Cuerpo

```
{
  "id": 1,
  "nombre": "Categoria 1",
  "descripcion": "Prueba",
  "categoria": {
    "id": 1,
    "nombre": "Categoria 1",
    "descripcion": "Prueba",
    "inventario": 1
  },
  "proveedor": {
    "id": 1,
    "nombre": "Proveedor 1",
    "descripcion": "Prueba"
  }
}
```

## #Lotes

### ## Gestion de Lotes

#### ### GET [/lotes/{id\_lote}]

Esta accion solo devuelve el registro de la tabla "Lote" con el "id\_lote" que se especifica y que este asociado al usuario logueado.

#### ### GET [/lotes]

Esta accion devuelve todos los Lotes que esten en la Base de Datos, según el usuario logueado.

+ Response 200 (application/json)

+ Atributos (object)

- + id: 1 (int, requerido)
- + costo: 10.00 (float)
- + precio: 15.00 (float)
- + existencia: 10 (int)

+ producto: 1 (int), Producto al que esta asociado el registro

+ Cuerpo

```
{
  "id": 1,
  "costo": 10.00
  "precio": 15.00
  "existencia": 10
  "producto": {
    "id": 1,
    "nombre": "Categoria 1" ,
    "descripcion": "Prueba",
    "categoria": 1
    "proveedor": 1
  }
}
```

### ### POST [/lotes]

Esta accion se encarga de crear un nuevo registro en la tabla de "lote".

+ Response 200 (application/json)

+ Atributos (object)

+ id: 1 (int, requerido)

+ costo: 10.00 (float)

+ precio: 15.00 (float)

+ existencia: 10 (int)

+ producto: 1 (int), Producto al que esta asociado el registro

+ Cuerpo

```
{
  "id": 1,
  "costo": 10.00
  "precio": 15.00
  "existencia": 10
  "producto": {
    "id": 1,
    "nombre": "Categoria 1" ,
```

```

        "descripcion": "Prueba",
        "categoria": 1
        "proveedor": 1
    }
}

```

### ### PUT [/lotes/{lotes\_id}]

Esta accion se encarga de modificar un registro en la tabla de "lote", según el "lote\_id" que se le mande como parametro.

+ Response 200 (application/json)

+ Atributos (object)

```

+ id: 1 (int, requerido)
+ costo: 10.00 (float)
+ precio: 15.00 (float)
+ existencia: 10 (int)
+ producto: 1 (int), Producto al que esta asociado el registro

```

+ Cuerpo

```

{
    "id": 1,
    "costo": 10.00
    "precio": 15.00
    "existencia": 10
    "producto": {
        "id": 1,
        "nombre": "Categoria 1",
        "descripcion": "Prueba",
        "categoria": 1
        "proveedor": 1
    }
}

```

### ### DELETE [/lotes/{lotes\_id}]

Esta accion se encarga de eliminar un registro en la tabla de "lote", según el "lote\_id" que se le mande como parametro.

+ Response 200 (application/json)

+ Cuerpo

```
{  
  "message": "OK"  
}
```