

## SERVICIOS HTTP REQUEST

A continuación se presenta el detalle de cada una de las interacciones identificadas hasta el momento, se utilizará JSON (no estricto) como formato para el intercambio de datos entre la aplicación iPhone y el servidor LAMP de Umni. En todas las peticiones se enviará un parámetro “idprovider” que contendrá un valor hexadecimal que identificará que las peticiones vienen de esta aplicación en concreto. El “idprovider” para esta aplicación es: **4xJTmc3\_umni\_es**

Todas las peticiones que pueden contener gran cantidad de registros en un futuro contienen un parámetro “start” y “limit” para segmentar o paginar el número de registros a cargar.

### TABLA DE CONTENIDOS:

GET .....	3
1 GETTOTALOFFERS .....	3
2 GETOFFERS .....	3
3 GETORDERS .....	5
4 GETORDERFOOD .....	8
5 GETFOODS .....	10
6 GETFOODOPTIONS .....	11
7 GETDAILYMENU .....	12
8 GETDAILYMENUFOOD .....	13
9 GETFOODCATEGORIES .....	14
10 GETRESTAURANTS .....	14
11 GETRESTAURANTTYPE .....	17
12 GETRESTAURANTPOSTALCODE .....	18
13 GETUSERMEMBERSHIP .....	18
14 GETUSER .....	19
15 GETUSERTOTALSPENDING .....	20
16 GETUSERTOTALSAVING .....	20
17 GETUSERRESTAURANTSVISITS .....	21
18 GETUSERADDRESS .....	21
19 VALIDATELOGIN .....	22
20 VALIDATEEMAIL .....	23

21	REMEMBERPASSWORD .....	24
22	SENDORDERTICKET .....	24
23	GETRESTAURANTSTARS .....	25
24	GETRESTAURANTSTARS AVG.....	25
25	VALIDATETABLENUMBER.....	26
26	GETIMAGE .....	26
27	GETFACEBOOKOFFERS .....	27
28	GETORDERSDEVOLUTION.....	28
29	GETHOURS OFFERS .....	29
30	GETDATES OFFERS .....	30
	SET .....	31
31	SETOFFER .....	31
32	SETORDER .....	32
33	SETORDERFOOD.....	34
34	SETFOOD.....	34
35	SETFOODGROUP .....	35
36	SETFOODOPTION .....	35
37	SETFOODCATEGORIES .....	36
38	SETDAILYMENU .....	36
39	SETDAILYMENUFOOD.....	37
40	SETRESTAURANT.....	37
41	SETRESTAURANTTYPE .....	39
42	SETMEMBERSHIP .....	39
43	SETUSERMEMBERSHIP .....	40
44	SETUSER .....	40
45	SETUSERTOTALSPENDING.....	41
46	SETUSERTOTALSAVING.....	41
47	SETUSERTOTALRESTAURANTSVISITS .....	42
48	SETUSERADDRESS.....	42
49	SETSUGGESTIONSPROBLEMS.....	43
50	SETRESTAURANTSTARSFAVORITES .....	43
51	DELETEUSER .....	44
52	DELETEUSERADDRESS.....	44

53	SETCHECKOUT .....	45
54	SETFACEBOOKOFFER.....	45

## GET

---

### 1 getTotalOffers

---

#### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, la respuesta contendrá el número total de ofertas que están disponibles en el saco.

#### Parámetros:

- iduser
- idprovider

#### URL petición:

*https://dominio.com/services/?a=gettotaloffer*

#### Formato de respuesta:

```
{ "gettotaloffer": {
  "offers": "<integer>",
}}
```

### 2 getOffers

---

#### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, y un parámetro "filter" que filtrará los datos devueltos de la siguiente manera:

0	Todos	2	Hoy (válidos hoy)
1	Favoritos		

Si se desea cambiar el orden se puede enviar en el campo order las siguientes opciones:

0	Últimos (últimos primero)	1	Cercanía (Más cercanos primero)
---	---------------------------	---	---------------------------------

En el caso de enviarse la opción 2 en el campo order deberá enviarse la posición actual del usuario en position.

El typediscount podrá ser:

0	Descuento sobre carta (%)	2	2x1 en categoría
1	Regalo de plato de Categoría (el de mayor valor)	3	Regalo de Plato (específico)

También en caso de querer filtrarse por restaurant se podrá enviar el idrestaurant.

La respuesta contendrá las ofertas que están disponibles en el saco, el id del restaurante al cuál enlaza, nombre del restaurante, nombre de la oferta, descripción, imagen, un campo position con las coordenadas GPS del restaurante de la oferta y un campo con la fecha de creación de la oferta.

Idfood contendrá el plato para los descuento de tipo 1 y 3, e idfoodCategories contendrá el id de la categoría para los descuentos de tipo 1 y 2, en el caso de discount contendrá el valor de los descuentos de tipo 0.

Adicionalmente la respuesta contendrá el parámetro favorite, que determinará si es uno de los marcados como favorito por el usuario (1=si, 0=no).

#### Parámetros:

- iduser
- filter (opcional)
- order (opcional)
- position (opcional)
- idrestaurant (opcional)
- idprovider
- start
- limit

#### URL petición:

*<https://dominio.com/services/?a=getoffers>*

#### Formato de respuesta:

```
{
  "getoffers": [
    {
      "idoffer": "<integer>",
      "idrestaurant": "<integer>",
      "namerestaurant": "<string>",
      "nameoffer": "<string>",
      "descriptionoffer": "<string>",
      "imageoffer": "<string Base64>",
      "position": "<string>",
      "discount": "<integer>",
      "idfood": "<integer>",
      "idfoodCategories": "<integer>",

```

```

    "typediscount": "<string>",
    "favorite": "<boolean>",
    "date_start": "<date>",
    "date_end": "<date>",
    "date": "<date>"
  }, {
    "idoffer": "<integer>",
    "idrestaurant": "<integer>",
    "namerestaurant": "<string>",
    "nameoffer": "<string>",
    "descriptionoffer": "<string>",
    "imageoffer": "<string Base64>",
    "position": "<string>",
    "discount": "<integer>",
    "idfood": "<integer>",
    "idfoodCategories": "<integer>",
    "typediscount": "<string>",
    "favorite": "<boolean>",
    "date_start": "<date>",
    "date_end": "<date>",
    "date": "<date>"
  }
  ...
  ...
  ...
}]

```

### 3 getOrders

---

#### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario. La respuesta contendrá las ordenes (o actividad) que el usuario tiene en su historial, ordenadas por fecha decreciente, si en caso contrario se desea solo una orden específica se podrá indicar el idorder.

Se incluye un campo status con el siguiente significado:

0	Pagado con tarjeta	3	Confirmado
1	Confirmado y Pagado con tarjeta	4	Pagado con efectivo
2	Confirmado y a pagar con efectivo	5	Pendiente de confirmación y pago
6	Confirmado y Pendiente de pago	7	Cobro fallido
8	Devolución efectuada	9	Devolución fallida
10	Devolución pendiente		

Un campo type con el siguiente significado:

1	Servicio a Domicilio	2	Servicio a Recoger
3	Servicio Antes de llegar al restaurante	4	Servicio En restaurante
5	Servicio de Reserva		

Un campo active con el siguiente significado:

0	Cerrado	1	Abierto
---	---------	---	---------

Un campo price con el coste del pedido, un campo persons con el número de personas (si se conoce), 4 campos de fecha que indican cuando se emitió el pedido (date), cuando se empezó a procesar (date\_start), cuando se terminó de procesar (date\_end), o para el caso de reservas la fecha futura de la misma (date\_reservation), adicionalmente se incluye un campo creditcard que indica si fue pagado con tarjeta de crédito (en caso afirmativo se incluyen los 4 últimos dígitos de la tarjeta, sino se encuentra vacío), un campo favorite en caso de marcar este pedido como favorito y un campo number\_table para los casos en que se haga el pedido en el restaurante y se indique el número de mesa.

El parámetro idorder\_food se incluye por si se desea pedir luego la información detallada de los platos de este pedido y sus costes (ver getOrderfood).

El parámetro idoffer se incluye en caso de que se haya aplicado una oferta a esta orden, por si se desea pedir luego la información detallada de la oferta y el descuento enviado el idorder y el idoffer (ver getOffer) el mismo caso sucedería para las ofertas de facebook.

Se añade el campo idorder\_tpv correspondiente al ID enviado para efectuar el cobro del pedido a través de la pasarela bancaria (TPV).

#### Parámetros:

- iduser
- idorder (opcional)
- type (opcional)
- idprovider
- start
- limit

#### URL petición:

*https://dominio.com/services/?a=getorders*

#### Formato de respuesta:

```
{
  "getorders": [
    {
```

```

    "idorder": "<integer>",
    "idorder_tpv": "<integer>",
    "idrestaurant": "<integer>",
    "idorder_food": "<integer>",
    "iduseraddress": "<integer>",
    "idoffer": "<integer>",
    "idoffer_facebook": "<integer>",
    "namerestaurant": "<string>",
    "status": "<integer>",
    "type": "<integer>",
    "instructions": "<string>",
    "persons": "<integer>",
    "date_start": "<date>",
    "date_end": "<date>",
    "date_reservation": "<date>",
    "date": "<date>",
    "creditcard": "<integer>",
    "favorite": "<boolean>",
    "number_table": "<integer>",
    "idoffer": "<integer>",
    "active": "<boolean>",
    "subtotal": "<decimal>",
    "membership_discount": "<decimal>",
    "offer_discount": "<decimal>",
    "facebook_discount": "<decimal>",
    "price_homedelivery": "<decimal>",
    "total": "<decimal>"
  }, {
    "idorder": "<integer>",
    "idrestaurant": "<integer>",
    "idorder_food": "<integer>",
    "iduseraddress": "<integer>",
    "namerestaurant": "<string>",
    "status": "<integer>",
    "type": "<integer>",
    "instructions": "<string>",
    "persons": "<integer>",
    "date_start": "<date>",
    "date_end": "<date>",
    "date_reservation": "<date>",
    "date": "<date>",
    "creditcard": "<integer>",
    "favorite": "<boolean>",
    "number_table": "<integer>",
    "idoffer": "<integer>",
    "active": "<boolean>",
    "subtotal": "<decimal>",
    "membership_discount": "<decimal>",

```

```

    "offer_discount": "<decimal>",
    "facebook_discount": "<decimal>",
    "price_homedelivery": "<decimal>",
    "total": "<decimal>"
  }
  ...
  ...
  ...
}]

```

## 4 getOrderfood

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario y el idorder. La respuesta contendrá la cantidad del plato (amount), nombre (namefood), descripción (descriptionfood), precio (price), tamaño (size), foto (imagefood) y en caso de pertenecer a un grupo de comida o pedirse dentro de un grupo (Ej. Un combo) se retornará el nombre del grupo (namefoodgroup), la descripción del grupo (descriptionfoodgroup) y el plus que se añade o quita al precio del plato (priceplusfoodgroup).

Igualmente retornará un parámetro type que en caso de indicarse platos de menú tendrá el siguiente significado:

0	Primero	2	Postre
1	Segundo	3	Bebida

En caso de haberse seleccionado una o más opciones (ej.: Con patatas) se devolverá un listado de las opciones con su id, el nombre y descripción de la opción y el precio que se añade o quita, y si no existe opción se devuelve vacío.

### Parámetros:

- iduser
- idorder
- idprovider
- start
- limit

### URL petición:

*<https://dominio.com/services/?a=getorderfood>*

### Formato de respuesta:

```

{"getorderfood": [
  {
    "idorder_food": "<integer>",

```



```

    "idfood": "<integer>",
    "iddailymenu_food": "<integer>",
    "amount": "<integer>",
    "namefood": "<string>",
    "descriptionfood": "<string>",
    "price": "<decimal>",
    "instructions": "<string>",
    "idfoodcategories": "<integer>",
    "imagefood": "<string Base64>",
    "idfoodgroup": "<integer>",
    "namefoodgroup": "<string>",
    "descriptionfoodgroup": "<string>",
    "priceplusfoodgroup": "<decimal>",
    "foodoptions": [
        {
            "idfoodoption": "<integer>",
            "namefoodoption": "<string>",
            "priceplusfoodoption": "<decimal>"
        }, {
            "idfoodoption": "<integer>",
            "namefoodoption": "<string>",
            "priceplusfoodoption": "<decimal>"
        }
        ...
        ...
        ...
    ]
}, {
    "idorder_food": "<integer>",
    "idfood": "<integer>",
    "iddailymenu_food": "<integer>",
    "amount": "<integer>",
    "namefood": "<string>",
    "descriptionfood": "<string>",
    "price": "<decimal>",
    "instructions": "<string>",
    "idfoodcategories": "<integer>",
    "imagefood": "<string Base64>",
    "idfoodgroup": "<integer>",
    "namefoodgroup": "<string>",
    "descriptionfoodgroup": "<string>",
    "priceplusfoodgroup": "<decimal>",
    "foodoptions": [
        {
            "idfoodoption": "<integer>",
            "namefoodoption": "<string>",
            "priceplusfoodoption": "<decimal>"
        }
    ]
}

```

```

    },{
      "idfoodoption": "<integer>",
      "namefoodoption": "<string>",
      "priceplusfoodoption": "<decimal>"
    }
    ...
    ...
    ...
  ]
}
...
...
...
}]

```

## 5 getFoods

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, el idrestaurant y el idfood, en caso de no enviarse el idfood se devuelven todos los platos, y en caso de especificarse menu (si contiene 1 es menu, si contiene 0 es carta), se devolverán solo los pertenecientes al menú del día.

La respuesta contendrá el nombre (namefood), descripción (descriptionfood), precio (price), foto (imagefood) y en caso de pertenecer a un grupo de comida o pedirse dentro de un grupo (Ej. Un combo) se retornará el nombre del grupo (namefoodgroup), la descripción del grupo (descriptionfoodgroup) y el plus que se añade o quita al precio del plato (priceplusfoodgroup).

Igualmente retornará un parámetro idfoodcategories y namefoodcategories que en caso de indicarse platos de menú contendrá el tipo o categoría del plato y su id.

En caso de existir opciones se devolverá un parámetro options a 1, y se encontrará en 0 en caso de no haber opciones. En caso de querer una lista de las opciones disponibles de un plato se puede utilizar getfoodoptions enviando el idfood.

### Parámetros:

- iduser
- idrestaurant
- idfood (opcional)
- menu
- idprovider
- start
- limit

**URL petición:**

*https://dominio.com/services/?a=getfoods*

**Formato de respuesta:**

```
{ "getfoods": [
  {
    "idfood": "<integer>",
    "namefood": "<string>",
    "descriptionfood": "<string>",
    "idfoodcategories": "<integer>",
    "namefoodcategories": "<string>",
    "price": "<decimal>",
    "imagefood": "<string Base64>",
    "idfoodgroup": "<integer>",
    "namefoodgroup": "<string>",
    "descriptionfoodgroup": "<string>",
    "priceplusfoodgroup": "<decimal>",
    "options": "<boolean>"
  }
  ...
  ...
  ...
]}
```

## 6 getFoodOptions

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario, el idrestaurant y el idfood.

Adicionalmente se debe enviar el parámetro obligatory que indicará si las opciones devueltas son las obligatorias o no, para el caso de que sean opciones pertenecientes a la sección "elije (obligatorias)" o a la opción "más ingredientes".

La respuesta contendrá el nombre (namefoodoption), descripción (descriptionfoodoption), y el precio (priceplusfoodoption).

**Parámetros:**

- iduser
- idrestaurant
- idfood
- obligatory

- idprovider
- start
- limit

**URL petición:**

*https://dominio.com/services/?a=getfoodoptions*

**Formato de respuesta:**

```
{ "getfoodoptions": [
  {
    "idfoodoption": "<integer>",
    "namefoodoption": "<string>",
    "descriptionfoodoption": "<string>",
    "priceplusfoodoption": "<decimal>"
  }, {
    "idfoodoption": "<integer>",
    "namefoodoption": "<string>",
    "descriptionfoodoption": "<string>",
    "priceplusfoodoption": "<decimal>"
  }
  ...
  ...
  ...
]}
```

## 7 getDailymenu

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario y el id del restaurante. La respuesta contendrá los menús disponibles en el restaurante, ordenados por precio decreciente.

**Parámetros:**

- iduser
- idrestaurant
- idprovider
- start
- limit

**URL petición:**

*https://dominio.com/services/?a=getdailymenu*

**Formato de respuesta:**

```
{ "getdailymenu": [
  {
```

```

    "iddailymenu": "<integer>",
    "price": "<decimal>",
    "date_start": "<date>",
    "date_end": "<date>",
    "date": "<date>"
  }, {
    "iddailymenu": "<integer>",
    "price": "<decimal>",
    "date_start": "<date>",
    "date_end": "<date>",
    "date": "<date>"
  }
  ...
  ...
  ...
}]

```

## 8 getDailymenufood

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario y el iddailymenu que se puede obtener con getDailymenu. La respuesta contendrá la lista de comida (idfood) y el nombre del tipo de comida de este menú, así como el id y el nombre del tipo de comida por si se necesitara agrupar.

### Parámetros:

- iduser
- iddailymenu
- idprovider
- start
- limit

### URL petición:

*<https://dominio.com/services/?a=getdailymenufood>*

### Formato de respuesta:

```

{"getdailymenufood": [
  {
    "iddailymenufood": "<integer>",
    "idfood": "<integer>",
    "idfoodcategories": "<integer>",
    "namefoodcategories": "<string>"
  }, {
    "iddailymenufood": "<integer>",
    "idfood": "<integer>",

```

```

        "idfoodcategories": "<integer>",
        "namefoodcategories": "<string>"
    }
    ...
    ...
    ...
  ]}

```

## 9 getFoodCategories

---

### Descripción:

Se enviará como parámetro mediante el método POST el id de la categoría en caso de querer obtener un nombre de categoría específica, de lo contrario si no se envía devuelve todas las categorías.

### Parámetros:

- idfoodcategories
- idprovider
- start
- limit

### URL petición:

*https://dominio.com/services/?a=getfoodcategories*

### Formato de respuesta:

```

{"getfoodcategories": [
  {
    "idfoodcategories": "<integer>",
    "namefoodcategories": "<string>"
  }, {
    "idfoodcategories": "<integer>",
    "namefoodcategories": "<string>"
  }
  ...
  ...
  ...
]}

```

## 10 getRestaurants

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, y

opcionalmente el idrestaurant, si este ultimo no se envía la respuesta contendrá toda la lista de restaurantes disponibles.

Como parámetro de envío también se incluye “type” que indica el tipo de servicio que debe tener el restaurante que se envía, si este viene vacio los devuelve todos:

1	Servicio a Domicilio	2	Servicio a Recoger
3	Servicio Antes de llegar al restaurante	4	Servicio En restaurante
5	Servicio de Reserva		

Si se desea cambiar el orden en que se listan los restaurantes se puede enviar en el campo order las siguientes opciones:

0	Más cercanos primero	1	Ofertas
2	Descuentos		

En el caso de enviarse la opción 0 en el campo order deberá enviarse la posición actual del usuario en position.

Si se desea filtrar el contenido se puede enviar en el campo idrestauranttype el id del tipo de restaurant, y/o en filter las siguientes opciones:

0	Todos	2	Histórico (los ya visitados)
1	Favoritos		

Si se desea filtrar por código postal, para saber si el servicio está disponible deberá indicar el iduseraddress del usuario que hace el pedido.

En la información recibida min\_price\_homedelivery contendrá el coste mínimo de la orden para realizar un envío a domicilio, y price\_homedelivery el coste de dicho envío.

Se incluyen 4 parámetros comenzados por “service\_” que indican los servicios que están disponibles, no disponibles o cerrados (por horario o fecha), estos parámetros devuelven los siguientes resultados:

0	No incluye este servicio	2	Esta cerrado o fuera de horario
1	Si incluye este servicio		

#### Parámetros:

- iduser
- idrestaurant
- idprovider
- iduseraddress (opcional)

- order (opcional)
- type (opcional)
- position (opcional)
- filter (opcional)
- idrestauranttype (opcional)
- start
- limit

**URL petición:**

*<https://dominio.com/services/?a=getrestaurants>*

**Formato de respuesta:**

```
{
  "getrestaurants": [
    {
      "idrestaurant": "<integer>",
      "name": "<string>",
      "description": "<string>",
      "price_average": "<integer>",
      "address1": "<string>",
      "address2": "<string>",
      "address3": "<string>",
      "position": "<string>",
      "idrestauranttype": "<integer>",
      "image1": "<string Base64>",
      "image2": "<string Base64>",
      "image3": "<string Base64>",
      "idoffer": "<integer>",
      "imageoffer": "<string Base64>",
      "min_price_homedelivery": "<decimal>",
      "price_homedelivery": "<decimal>",
      "service_adomicilio": "<integer>",
      "service_arecoger": "<integer>",
      "service_antesrestaurante": "<integer>",
      "service_enrestaurante": "<integer>",
      "visa": "<integer>",
      "mastercard": "<integer>",
      "amex": "<integer>",
      "date": "<date>"
    }, {
      "idrestaurant": "<integer>",
      "name": "<string>",
      "description": "<string>",
      "price_average": "<integer>",
      "address1": "<string>",
      "address2": "<string>",
      "address3": "<string>",
      "position": "<string>",
      "idrestauranttype": "<integer>"
    }
  ]
}
```



```

    "image1": "<string Base64>",
    "image2": "<string Base64>",
    "image3": "<string Base64>",
    "min_price_homedelivery": "<decimal>",
    "price_homedelivery": "<decimal>",
    "service_adomicilio": "<integer>",
    "service_arecoger": "<integer>",
    "service_antesrestaurante": "<integer>",
    "service_enrestaurante": "<integer>",
    "visa": "<integer>",
    "mastercard": "<integer>",
    "amex": "<integer>",
    "date": "<date>"
  }
  ...
  ...
  ...
  ]}

```

## 11 getRestauranttype

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, y el idrestauranttype, si este ultimo no se envía la respuesta contendrá toda la lista de tipos de restaurantes (Comida Italiana, Parrillada Argentina, etc), así mismo si se especifica un idrestaurant se devuelve solo el o los tipos de comida del restaurant específico.

### Parámetros:

- iduser
- idrestauranttype (opcional)
- idrestaurant (opcional)
- idprovider
- start
- limit

### URL petición:

*https://dominio.com/services/?a=getrestauranttype*

### Formato de respuesta:

```

{"getrestauranttype": [
  {
    "idrestauranttype": "<integer>",
    "name": "<string>",
  }, {

```

```

        "idrestauranttype": "<integer>",
        "name": "<string>",
    }
    ...
    ...
    ...
}]

```

## 12 getRestaurantpostalcode

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del restaurant y el iduseraddress, y recibe como respuesta si el restaurant da servicio a domicilio en el código postal que indicó el usuario en su dirección (1=si, 0=no).

### Mensaje en caso de devolver "0":

"Lo sentimos, el servicio a domicilio del restaurante seleccionado no esta disponible en su zona."

### Parámetros:

- iduser
- idrestaurant
- iduseraddress
- idprovider

### URL petición:

*<https://dominio.com/services/?a=getrestaurantpostalcode>*

### Formato de respuesta:

```

{"getrestaurantpostalcode":{
    "result": "<boolean>",
}}

```

## 13 getUsermembership

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, y el idrestaurant, la respuesta contendrá el nombre de la membresía o estado que tiene el usuario (Ej.: Gold), la descripción del estado en que se encuentra, el dinero acumulado y el beneficio que obtiene en esta membresía.

Adicionalmente se incluye el nombre de la membresía anterior y siguiente a donde se encuentra ubicado el usuario, y el precio limite de la membresía anterior y siguiente para poder ubicar el punto exacto donde se encuentra el usuario en el intervalo.

**Parámetros:**

- iduser
- idrestaurant
- idprovider
- start
- limit

**URL petición:**

*https://dominio.com/services/?a=getusermembership*

**Formato de respuesta:**

```
{ "getusermembership": [
  {
    "idmembership": "<integer>",
    "membershipname": "<string>",
    "membershippricemax": "<string>",
    "membershippricemin": "<string>",
    "membershipdescription": "<string>",
    "priceaccumulated": "<decimal>",
  }
]}
```

## 14 getUser

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario, y la respuesta contendrá todos los datos del usuario, incluido total\_spending que contendrá el gasto total hecho por el usuario en todas sus ordenes, total\_saving que contendrá el ahorro total que suma el usuario, y restaurants\_visits que contiene el número de restaurantes donde ha gastado y ahorrado lo anterior.

**Parámetros:**

- iduser
- idprovider

**URL petición:**

*https://dominio.com/services/?a=getuser*

**Formato de respuesta:**

```
{ "getuser": {
  "iduser": "<integer>",
  "name": "<string>",

```

```

    "lastname": "<string>",
    "email": "<string>",
    "location": "<string>",
    "phone": "<integer>",
    "genre": "<tinyint>",
    "birthday": "<date>",
    "total_spending": "<decimal>",
    "total_saving": "<decimal>",
    "restaurants_visits": "<integer>",
    "type": "<integer>"
    "date": "<date>"
  }}

```

## 15 getUsertotalspending

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, y la respuesta contendrá total\_spending con el gasto total hecho por el usuario en todas sus ordenes.

### Parámetros:

- iduser
- idprovider

### URL petición:

*https://dominio.com/services/?a=getusertotalspending*

### Formato de respuesta:

```

{"getusertotalspending": {
  "total_spending": "<decimal>"
}}

```

## 16 getUsertotalsaving

---

### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario, y la respuesta contendrá total\_saving que contendrá el ahorro total que suma el usuario.

### Parámetros:

- iduser
- idprovider

**URL petición:**

*https://dominio.com/services/?a=getusertotalsaving*

**Formato de respuesta:**

```
{ "getusertotalsaving": {  
    "total_saving": "<decimal>"  
}}
```

## 17 getUserrestaurantsvisits

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario, y la respuesta contendrá restaurants\_visits que contiene el número de restaurantes que ha visitado.

**Parámetros:**

- iduser
- idprovider

**URL petición:**

*https://dominio.com/services/?a=getuserrestaurantsvisits*

**Formato de respuesta:**

```
{ "getuserrestaurantsvisits": {  
    "restaurants_visits": "<integer>"  
}}
```

## 18 getUseraddress

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario, y la respuesta contendrá toda la lista de direcciones de este usuario, incluido el name que la identifica (casa, trabajo, etc.).

**Parámetros:**

- iduser
- idprovider
- start
- limit

**URL petición:**

*https://dominio.com/services/?a=getuseraddress*

### Formato de respuesta:

```
{ "getuseraddress": [
  {
    "iduseraddress": "<integer>",
    "name": "<string>",
    "address1": "<string>",
    "address2": "<string>",
    "address3": "<string>",
    "address4": "<string>",
    "cp": "<integer>",
    "location": "<string>",
    "phone": "<integer>",
    "mobile": "<integer>"
  }, {
    "iduseraddress": "<integer>",
    "name": "<string>",
    "address1": "<string>",
    "address2": "<string>",
    "address3": "<string>",
    "address4": "<string>",
    "cp": "<integer>",
    "location": "<string>",
    "phone": "<integer>",
    "mobile": "<integer>"
  }
  ...
  ...
  ...
]}
```

## 19 validateLogin

---

### Descripción:

Se enviará como parámetro mediante el método POST el usuario, clave y el token la respuesta contendrá todos los datos del usuario en caso de ser correcto, de lo contrario contendrá resultado "0" en caso de ser incorrecto y "-1" en caso de estar bloqueado o "baneado", y además el mensaje de respuesta.

### Parámetros:

- email
- password
- idprovider
- token

### URL petición:

*https://dominio.com/services/?a=validatelogin*

### **Formato de respuesta en caso de ser incorrecto o estar bloqueado**

```
{ "validatelogin": {
  "result": "<boolean>",
  "msg": "<string>"
}}
```

### **Formato de respuesta en caso de ser correcto:**

```
{ "validatelogin": {
  "iduser": "<integer>",
  "name": "<string>",
  "lastname": "<string>",
  "email": "<string>",
  "location": "<string>",
  "phone": "<integer>",
  "genre": "<tinyint>",
  "birthday": "<date>",
  "total_spending": "<decimal>",
  "total_saving": "<decimal>",
  "restaurants_visits": "<integer>",
  "type": "<integer>",
  "date": "<date>"
}}
```

## **20 validateEmail**

---

### **Descripción:**

Se enviará como parámetro mediante el método POST el correo con que se desea suscribir, el token del dispositivo que se esta registrando y desde que tipo de suscripción se hace la petición (1 para facebook y 0 para el resto de peticiones), la respuesta contendrá el mensaje de respuesta (en caso de que sea necesario) y el resultado:

- iduser : en caso de que exista previamente, se haya registrado vía Facebook y se envíe type con valor igual a "1".
- 1: En cualquier caso distinto al anterior donde el correo ya este registrado.
- -1 : En caso de que el usuario este bloqueado o "baneado".
- 0 : en caso de que no exista este correo.

### **Parámetros:**

- email
- idprovider
- type
- token

**URL petición:**

*https://dominio.com/services/?a=validateemail*

**Formato de respuesta:**

```
{"validateemail":{  
  "result": "<integer>",  
  "msg": "<string>"  
}}
```

## 21 rememberPassword

---

**Descripción:**

Se enviará como parámetro mediante el método POST el correo electrónico, esto generará el envío de una nueva clave vía correo electrónico, la respuesta contendrá 1 en caso de ser enviada o 0 en caso de no ser enviada a causa de un error.

**Mensaje en caso de error:**

"Lo sentimos, ha ocurrido un error, por favor inténtelo nuevamente".

**Parámetros:**

- email
- idprovider

**URL petición:**

*https://dominio.com/services/?a=rememberpassword*

**Formato de respuesta:**

```
{"rememberpassword":{  
  "result": "<boolean>"  
}}
```

## 22 sendOrderticket

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario y el idorder, esto generará el envío del ticket vía correo electrónico, la respuesta



contendrá 1 en caso de ser enviada o 0 en caso de no ser enviada a causa de un error.

**Mensaje en caso de error:**

“Lo sentimos, ha ocurrido un error al enviar el pedido, por favor inténtelo nuevamente”.

**Parámetros:**

- iduser
- idorder
- idprovider

**URL petición:**

*https://dominio.com/services/?a=sendorderticket*

**Formato de respuesta:**

```
{ "sendorderticket": {  
  "result": "<boolean>"  
}}
```

## 23 getRestaurantStars

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario y el id del restaurante, y la respuesta contendrá el número de estrellas que le ha dado el usuario a ese restaurante.

**Parámetros:**

- iduser
- idrestaurant
- idprovider

**URL petición:**

*https://dominio.com/services/?a=getrestaurantsstars*

**Formato de respuesta:**

```
{ "getrestaurantsstars": {  
  "result": "<integer>"  
}}
```

## 24 getRestaurantStarsAVG

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del restaurante, y la respuesta contendrá el número promedio de estrellas que le han dado todos los usuarios a ese restaurante.

**Parámetros:**

- idrestaurant
- idprovider

**URL petición:**

*https://dominio.com/services/?a=getrestaurantsstarsavg*

**Formato de respuesta:**

```
{ "getrestaurantsstarsavg": {  
  "result": "<integer>"  
}}
```

## 25 validateTablenumber

---

**Descripción:**

Se enviará como parámetro mediante el método POST el idrestaurant con el código (code) de la mesa que se desea validar, la respuesta contendrá 1 en caso de ser correcto o 0 en caso de existir previamente.

**Mensaje en caso de devolver "0":**

"Lo sentimos, este número de mesa no existe en este restaurante, verifique la información e inténtelo nuevamente."

**Parámetros:**

- idrestaurant
- code
- idprovider

**URL petición:**

*https://dominio.com/services/?a=validatetablename*

**Formato de respuesta:**

```
{ "validatetablename": {  
  "result": "<boolean>,"  
}}
```

## 26 getImage

---

### Descripción:

Permite cargar cualquier tipo de image directamente desde una URL. Se enviará como parámetro mediante el método POST el id del usuario, un id que identifica el contenido y el type con el tipo de contenido, adicionalmente se enviará los parámetros “w” y “h” con las medidas (en pixeles) a las que se convertirá la imagen dinámicamente conservando las proporciones.

El parámetro type podrá ser el siguiente:

1	Restaurants	2	Offers
3	Foods		

Cada uno de los tipos anteriores devuelve una sola imagen, para el caso específico de Restaurants se debe enviar un parámetro adicional, el cual es img\_number, que contendrá un valor numérico del 1 al 9, que devolvera las 9 fotos que puede tener un restaurante.

### Parámetros:

- iduser
- id
- type
- img\_number (opcional)
- w
- h
- idprovider

### URL petición:

*https://dominio.com/services/?a=getimage*

### Formato de respuesta:

Imagen en formato JPEG.

## 27 getFacebookoffers

---

### Descripción:

Permite cargar la oferta de facebook única por restaurant. Se enviará como parámetro mediante el método POST el id del usuario y el id del restaurante, y el tipo de orden, la respuesta contendrá offer\_description que aparecerá en el recuadro de la App, facebook\_description que aparecerá en facebook, idfood que

será el ítem que se regalará y idoffer\_facebook que se necesitará posteriormente para indicar cuando la oferta sea utilizada.

**Importante:** En caso de que no haya oferta en un restaurante este espacio para una oferta de facebook se utilizará como un “recomienda” estándar igual que el otro existente y al pulsar se guardará con el método setRestaurantStarsfavorites.

En el campo type se debe enviar uno de los siguientes:

1	Servicio a Domicilio	2	Servicio a Recoger
3	Servicio Antes de llegar al restaurante	4	Servicio En restaurante
5	Servicio de Reserva		

#### Parámetros:

- iduser
- idrestaurant
- idprovider
- type
- idoffer\_facebook (opcional)

#### URL petición:

*https://dominio.com/services/?a=getfacebookoffers*

#### Formato de respuesta:

```
{ "getfacebookoffers": {
  "idoffer_facebook": "<integer>",
  "offer_description": "<string>",
  "facebook_description": "<string>",
  "idfood": "<integer>"
}}
```

## 28 getOrdersDevolution

---

#### Descripción:

Se enviará como parámetro mediante el método POST el id del usuario. La respuesta contendrá la o los pedidos para las cuáles hay que tramitar la devolución del pago a través de la pasarela bancaria (TPV).

#### Parámetros:

- iduser
- idprovider

**URL petición:**

*https://dominio.com/services/?a=getordersdevolution*

**Formato de respuesta:**

```
{ "getordersdevolution": [
  {
    "idorder": "<integer>",
    "idorder_tpv": "<integer>",
    "idrestaurant": "<integer>",
    "namerestaurant": "<string>",
    "date_reservation": "<date>",
    "date": "<date>",
    "creditcard": "<integer>",
    "type_creditcard": "<string>",
    "subtotal": "<decimal>",
    "membership_discount": "<decimal>",
    "offer_discount": "<decimal>",
    "facebook_discount": "<decimal>",
    "price_homedelivery": "<decimal>",
    "total": "<decimal>"
  }, {
    "idorder": "<integer>",
    "idorder_tpv": "<integer>",
    "idrestaurant": "<integer>",
    "namerestaurant": "<string>",
    "date_reservation": "<date>",
    "date": "<date>",
    "creditcard": "<integer>",
    "type_creditcard": "<string>",
    "subtotal": "<decimal>",
    "membership_discount": "<decimal>",
    "offer_discount": "<decimal>",
    "facebook_discount": "<decimal>",
    "price_homedelivery": "<decimal>",
    "total": "<decimal>"
  }
  ...
  ...
  ...
]}
```

## 29 getHoursOffers

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario y el id del

restaurante donde desea realizar una reserva o pedido con reserva. La respuesta contendrá la horas disponibles para reservar en ese restaurante, indicando si cada una de las horas tiene ofertas disponibles. Hay que acotar que solo contendrá las horas disponibles no todo el rango de horas del día.

**Parámetros:**

- iduser
- idprovider
- idrestaurant
- date (yyyy-mm-dd)

**URL petición:**

*https://dominio.com/services/?a=gethoursoffers*

**Formato de respuesta:**

```
{ "gethoursoffers": [
  {
    "<time>": "<integer>"
  }, {
    "<time>": "<integer>"
  }
  ...
  ...
  ...
]}
```

## 30 getDatesOffers

---

**Descripción:**

Se enviará como parámetro mediante el método POST el id del usuario y el id del restaurante donde desea realizar una reserva o pedido con reserva. La respuesta contendrá la fechas que tienen ofertas disponibles. Hay que acotar que solo contendrá las fechas con ofertas disponibles no todo el rango de fechas del mes, y solo contendrá el mes en curso que se consulta.

**Parámetros:**

- iduser
- idprovider
- idrestaurant
- month (mm)

**URL petición:**

*https://dominio.com/services/?a=getdatesoffers*

**Formato de respuesta:**

```

{"getdatesoffers":[
  {
    "<date>":"<integer>"
  },{
    "<date>":"<integer>"
  }
  ...
  ...
  ...
]}

```

## SET

---

### 31 setOffer

---

#### Descripción:

Se enviará como parámetros mediante el método POST los siguientes datos:

idoffer (integer) (opcional)  
 idrestaurant (integer)  
 nameoffer (string)  
 descriptionoffer (string)  
 imageoffer (image)  
 discount (integer)  
 idfood (integer)  
 idfoodCategories (integer)  
 typediscount (integer)  
 date\_start (date)  
 date\_end (date)

Los datos deben contener el id del restaurante, nombre de la oferta, descripción, imagen, el descuento, tipo de descuento, fecha de comienzo y fin de la oferta.

El typediscount podrá ser:

0	Descuento sobre carta (%)	2	2x1 en categoría
1	Regalo de plato de Categoría (el de mayor valor)	3	Regalo de Plato (específico)

Idfood contendrá el plato para los descuento de tipo 1 y 3, e idfoodCategories contendrá el id de la categoría para los descuentos de tipo 1 y 2, en el caso de discount contendrá el valor de los descuentos de tipo 0.

En caso de incluirse el idoffer se considerará una actualización de una oferta anterior.

**URL petición:**

*https://dominio.com/services/?a=setoffer*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idoffer> en caso contrario "0".

## 32 setOrder

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

idorder (integer) (opcional)  
 idorder\_tpv (integer) (opcional)  
 idrestaurant (integer) (opcional)  
 iduser (integer)  
 iduserAddress (integer) (opcional)  
 idoffer (integer) (opcional)  
 idoffer\_facebook (integer) (opcional)  
 persons (integer) (opcional)  
 instructions (string) (opcional)  
 date\_reservation (date) (opcional)  
 number\_creditcard (integer) (opcional)  
 type\_creditcard (string) (opcional)  
 favorite (boolean) (opcional)  
 type (integer) (opcional)  
 number\_table (integer) (opcional)  
 status (integer) (opcional)  
 idoffer (integer) (opcional)  
 active (boolean) (opcional)  
 subtotal (decimal) (opcional)  
 membership\_discount (decimal) (opcional)  
 offer\_discount (decimal) (opcional)  
 facebook\_discount (decimal) (opcional)  
 price\_homedelivery (decimal) (opcional)  
 total (decimal) (opcional)

Se incluye un campo type con el siguiente significado:

1	Servicio a Domicilio	2	Servicio a Recoger
3	Servicio Antes de Llegar al restaurante	4	Servicio En restaurante
5	Servicio de Reserva		



un campo status con el siguiente significado:

0	Pagado con tarjeta	3	Confirmado
1	Confirmado y Pagado con tarjeta	4	Pagado con efectivo
2	Confirmado y a pagar con efectivo	5	Pendiente de confirmación y pago
6	Confirmado y Pendiente de pago	7	Cobro fallido
8	Devolución efectuada	9	Devolución fallida
10	Devolución pendiente		

Un campo active con el siguiente significado:

0	Cerrado	1	Abierto
---	---------	---	---------

El campo persons con el número de personas (si se conoce), un campo de fecha que indica para cuando se reserva la mesa (date\_reservation), adicionalmente un campo number\_table para los casos en que se haga el pedido en el restaurante y se indique el número de mesa.

El campo type\_creditcard deberá ser "visa" o "mastercard".

#### **Actualización de ordenes:**

- Todos los parámetros son opcionales ya que se puede actualizar casi cualquier aspecto de una orden, solo enviando el idorder y el, o los, parámetros que se quieran actualizar.
- Solo hay que tener presente que es imprescindible que se envíe adicionalmente idrestaurant e iduser si se desea actualizar el total de la orden.
- Las actualizaciones de totales, subtotales, descuentos por membresía, oferta o facebook deben contener siempre la diferencia, tanto en negativo como en positivo con respecto a la orden anterior, y nunca un nuevo valor pensado de forma sustitutiva del anterior.

#### **URL petición:**

*<https://dominio.com/services/?a=setorder>*

#### **Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idorder> en caso contrario "0".

#### **Mensaje en caso de devolver "0":**

"Lo sentimos, ha ocurrido un error al intentar enviar su pedido, por favor inténtelo nuevamente pasados unos minutos."

## 33 setOrderfood

---

### Descripción:

Se enviará como parámetros mediante el método POST los siguientes datos:

idorder (integer)  
idfood (integer)  
iddailymenufood (integer) (opcional)  
idfoodoptions (string) (opcional)  
idfoodgroup (string) (opcional)  
amount (integer)  
instructions (string) (opcional)

Se enviará el idfood y el idorder a la que pertenece, la cantidad del plato (amount), y en caso de pertenecer a un grupo de comida o pedirse dentro de un grupo se debe enviar el idfoodgroup. Igualmente se indicará el iddailymenu\_food si pertenece a un plato de menú o se dejará vacío en caso contrario, y en idfoodoptions se indicarán los id de las opciones elegidas para ese plato separadas por coma sin espacios.

### URL petición:

*<https://dominio.com/services/?a=setorderfood>*

### Formato de respuesta:

En caso de realizarse exitosamente la operación se devuelve <idorderfood> en caso contrario "0".

### Mensaje en caso de devolver "0":

"Lo sentimos, ha ocurrido un error al intentar enviar su pedido, por favor inténtelo nuevamente."

## 34 setFood

---

### Descripción:

Se enviará como parámetros mediante el método POST los siguientes datos:

idfood (integer) (opcional)  
namefood (string)  
descriptionfood (string)  
price (decimal)  
idfoodgroup (boolean)  
imagefood (image)  
idfoodcategories (integer)

Se enviará el nombre (namefood), descripción (descriptionfood), precio (price), foto (imagefood) y en caso de pertenecer a un grupo de comida o pedirse dentro de un grupo (Ej. Un combo) se enviará un parámetro idfoodgroup con el id del grupo al que pertenece.

Igualmente se envía un parámetro idfoodcategories que identifica la categoría a la que pertenece.

En caso de incluirse el idfood se considerará una actualización de una comida.

**URL petición:**

*<https://dominio.com/services/?a=setfood>*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idfood> en caso contrario "0".

## 35 setFoodgroup

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

idfoodgroup (integer) (opcional)  
namefoodgroup (string)  
descriptionfoodgroup (string)  
priceplusfoodgroup (decimal)

En caso de incluirse el idfoodgroup se considerará una actualización de un grupo de comidas.

**URL petición:**

*<https://dominio.com/services/?a=setfoodgroup>*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idfoodgroup> en caso contrario "0".

## 36 setFoodOption

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

idfoodoption (integer) (opcional)

idfood (integer)  
namefoodoption (string)  
descriptionfoodoption (string)  
priceplusfoodoption (decimal)  
obligatory (boolean)

Se envía el parámetro obligatory que indicará si la opción enviada pertenece a una de las opciones obligatorias pertenecientes a la sección “elije (obligatorias)” o en caso contrario a la opción “más ingredientes”.

En caso de incluirse el idfoodoption se considerará una actualización de un grupo de comidas.

**URL petición:**

*https://dominio.com/services/?a=setfoodoption*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idfoodoption> en caso contrario “0”.

## 37 setFoodCategories

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

idfoodcategories (integer) (opcional)  
name (string)

En caso de incluirse el idfoodcategories se considerará una actualización de la categoría.

**URL petición:**

*https://dominio.com/services/?a=setfoodcategories*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idfoodcategories> en caso contrario “0”.

## 38 setDailymenu

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

iddailymenu (integer) (opcional)

price (decimal)  
date\_start (date)  
date\_end (date)

En caso de incluirse el iddailymenu se considerará una actualización del menú del día.

**URL petición:**

*https://dominio.com/services/?a=setdailymenu*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iddailymenu> en caso contrario "0".

## 39 setDailymenufood

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

iddailymenufood (integer) (opcional)  
iddailymenu (integer)  
idfood (integer)  
idfoodcategories (integer)

Los tipos o categorías de comidas (idfoodcategories) se obtienen con el método getFoodCategories

En caso de incluirse el iddailymenufood se considerará una actualización de los platos del menú del día.

**URL petición:**

*https://dominio.com/services/?a=setdailymenufood*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iddailymenufood> en caso contrario "0".

## 40 setRestaurant

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

idrestaurant (integer) (opcional)  
name (string)

description (string)  
price\_average (integer)  
address1 (string)  
address2 (string)  
address3 (string)  
position (string)  
idrestauranttype (integer)  
image1 (image)  
image2 (image)  
image3 (image)  
min\_price\_homedelivery (decimal)  
price\_homedelivery (decimal)  
service\_adomicilio (integer)  
service\_arecoger (integer)  
service\_antesrestaurante (integer)  
service\_enrestaurante (integer)  
service\_reserva (integer)  
service\_adomicilio\_hours\_start (string)  
service\_adomicilio\_hours\_end (string)  
service\_arecoger\_hours\_start (string)  
service\_arecoger\_hours\_end (string)  
service\_antesrestaurante\_hours\_start (string)  
service\_antesrestaurante\_hours\_end (string)  
days\_closed (string)  
visa (integer)  
mastercard (integer)  
amex (integer)

min\_price\_homedelivery contendrá el coste mínimo de la orden para realizar un envío a domicilio, y price\_homedelivery el coste de dicho envío.

Los parámetros comenzamos “con service\_” recogerán los servicios activos e inactivos por restaurant, además de los horarios de los 7 días de la semana separado por comas, y en otro parámetro days\_closed los días del año en que permanecen cerrados.

En caso de incluirse el idrestaurant se considerará una actualización de los datos del restaurante.

**URL petición:**

*<https://dominio.com/services/?a=getrestaurant>*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idrestaurant> en caso contrario “0”.

## 41 setRestauranttype

---

### Descripción:

Se enviará como parámetros mediante el método POST los siguientes datos:

idrestauranttype (integer) (opcional)  
 name (string)  
 description (string)

En caso de incluirse el idrestauranttype se considerará una actualización de los datos del restaurante.

### URL petición:

*https://dominio.com/services/?a=getrestauranttype*

### Formato de respuesta:

En caso de realizarse exitosamente la operación se devuelve <idrestauranttype> en caso contrario "0".

## 42 setMembership

---

### Descripción:

Se enviará como parámetros mediante el método POST los siguientes datos:

idmembership (integer) (opcional)  
 idrestaurant (integer)  
 namemembership (string)  
 descriptionmembership (string)  
 pricemax (decimal)  
 pricemin (decimal)  
 value\_accumulated (integer)  
 type\_accumulated (integer)

Los tipos (type\_accumulated) de membresías se dividen en:

0	Cantidad de euros del consumo	1	Porcentaje de Euros del consumo
---	-------------------------------	---	---------------------------------

Se acumula una cantidad (value\_accumulated) de un tipo (type\_accumulated), en caso de que dicha cantidad se sitúe entre pricemin y pricemax.

En caso de incluirse el idmembership se considerará una actualización de los valores de la membresía.

### URL petición:

*https://dominio.com/services/?a=setmembership*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <idmembership> en caso contrario "0".

## 43 setUsermembership

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

idrestaurant (integer)  
iduser (integer)  
idmembership (decimal)  
price (decimal)

**URL petición:**

*https://dominio.com/services/?a=setusermembership*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iduser\_restaurants\_memberships> en caso contrario "0".

## 44 setUser

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

iduser (integer) (opcional)  
name (string)  
lastname (string)  
email (string)  
password (string)  
location (string) (opcional)  
phone (integer) (opcional)  
genre (tinyint) (opcional)  
birthday (date) (opcional)  
type (integer)  
token (string)  
friends (integer)

En type se indicará el tipo de alta: 0 (normal), 1 (Facebook). En friends se indicará el número de amigos que tiene en Facebook.



**URL petición:**

*https://dominio.com/services/?a=setuser*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iduser> en caso contrario "0".

**Mensaje en caso de devolver "0":**

"Lo sentimos, ha ocurrido un error al intentar crear su usuario, por favor inténtelo nuevamente pasados unos minutos."

## 45 setUserTotalSpending

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

iduser (integer)

total\_spending (decimal)

**URL petición:**

*https://dominio.com/services/?a=setUserTotalSpending*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iduser> en caso contrario "0".

## 46 setUserTotalSaving

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

iduser (integer)

total\_saving (decimal)

**URL petición:**

*https://dominio.com/services/?a=setUserTotalSaving*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iduser> en caso contrario "0".

## 47 setUserTotalRestaurantsVisits

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

iduser (integer)  
restaurants\_visits (integer)

**URL petición:**

*https://dominio.com/services/?a=setUserTotalRestaurantsVisits*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iduser> en caso contrario "0".

## 48 setUserAddress

---

**Descripción:**

Se enviará como parámetros mediante el método POST los siguientes datos:

iduseraddress (integer) (opcional)  
iduser (integer)  
name (string)  
address1 (string)  
address2 (string) (opcional)  
address3 (string) (opcional)  
address4 (string) (opcional)  
address5 (string) (opcional)  
address6 (string) (opcional)  
cp (integer)  
location (string)  
phone (integer) (opcional)  
mobile (integer)

**URL petición:**

*https://dominio.com/services/?a=setUserAddress*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve <iduseraddress> en caso contrario "0".

**Mensaje en caso de devolver "0":**

“Lo sentimos, ha ocurrido un error al intentar crear una nueva dirección, por favor inténtelo nuevamente pasados unos minutos.”

## 49 setSuggestionsproblems

---

### Descripción:

Se enviará como parámetros mediante el método POST el iduser, description que contendrá la descripción del problema o sugerencia, y type que será 0 para sugerencias y 1 para problemas.

iduser (integer)  
description (string)  
type (integer)

### URL petición:

*https://dominio.com/services/?a=setsuggestionsproblems*

### Formato de respuesta:

En caso de realizarse exitosamente la operación se devuelve 1 en caso contrario “0”.

### Mensaje en caso de devolver “0”:

“Lo sentimos, ha ocurrido un error al intentar enviar el formulario, por favor inténtelo nuevamente pasados unos minutos.”

## 50 setRestaurantStarsfavorites

---

### Descripción:

Se enviará como parámetros mediante el método POST el iduser e idrestaurant, y además stars o favorites según el caso, y type que especifica lo que se está enviando:

0	favoritos	3	Me Gusta (Recomienda)
1	Stars		

El campo favorite deberá contener un valor 1 (si) o 0 (no), el campo stars puede contener valores del 1 al 5 según la valoración del restaurante que dé el usuario, y para el caso de Me Gusta no se necesita enviar un valor.

iduser (integer)  
idrestaurant (integer)  
favorite (integer)  
stars (integer)

type (integer)

**URL petición:**

*https://dominio.com/services/?a=setrestaurantstarsfavorites*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve 1 en caso contrario "0".

**Mensaje en caso de devolver "0":**

"Lo sentimos, ha ocurrido un error al intentar guardar los datos, por favor inténtelo nuevamente pasados unos minutos."

## 51 deleteUser

---

**Descripción:**

Se enviará como parámetros mediante el método POST el iduser.

iduser (integer)

**URL petición:**

*https://dominio.com/services/?a=deleteuser*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve 1 en caso contrario "0".

**Mensaje en caso de devolver "0":**

"Lo sentimos, ha ocurrido un error al intentar eliminar este usuario, por favor inténtelo nuevamente pasados unos minutos."

## 52 deleteUserAddress

---

**Descripción:**

Se enviará como parámetros mediante el método POST el iduseraddress

iduseraddress (integer)

**URL petición:**

*https://dominio.com/services/?a=deleteuseraddress*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve 1 en caso contrario "0".

## 53 setCheckout

---

### Descripción:

Registra si alguien a pedido la cuenta o ha pagado una orden ya emitida. Se enviará como parámetros mediante el método POST los siguientes datos:

idprovider (integer)  
 idorder (integer)  
 type (integer)  
 number\_creditcard (integer) (opcional)  
 type\_creditcard (string) (opcional)

Los tipos (type) se dividen en:

0	Pago con tarjeta	1	Solicitud de cuenta (pago con efectivo)
---	------------------	---	---

Este proceso actualiza automáticamente el status de la orden, en el primer caso se cambia a 1 (confirmado y pagado con tarjeta), y en el segundo a 2 (confirmado y pagado con efectivo)

### URL petición:

*https://dominio.com/services/?a=setcheckout*

### Formato de respuesta:

En caso de realizarse exitosamente la operación se devuelve "1" en caso contrario "0".

## 54 setFacebookoffer

---

### Descripción:

Registra que alguien ha pulsado la oferta de facebook. Se incluye el parámetro "future" que se utilizará en el caso de que sea una oferta que se aplicará a futuro (1=si, 0=no). Se enviará como parámetros mediante el método POST los siguientes datos:

idprovider (integer)  
 idoffer\_facebook (integer)  
 iduser (integer)  
 future (boolean)

### URL petición:

*https://dominio.com/services/?a=setfacebookoffer*

**Formato de respuesta:**

En caso de realizarse exitosamente la operación se devuelve “1” en caso contrario “0”.