*MARKETPLACE-EOI*

*‘Grupo G2’*

**Fecha Entrega: Mayo 2021**

**Integrantes:**

**Manuel Herrera González**

**David Hernández Miranzo**

*©:Quedan reservados todos los derechos. (Ley de Propiedad Intelectual del 17 de noviembre de 1987 y Reales Decretos).   
Documentación elaborada por la EOI.  
Prohibida la reproducción total o parcial sin autorización escrita de la EOI.*

1. **Memoria Descriptiva**
   1. ***Organización***

**Equipo**

David Hernández Miranzo (Portavoz)

Manuel Herrera González.

**Distribución del trabajo**

David Hernández Miranzo:

* Dashboard
* Login
* Nav
* Servicios
* CSS

Manuel Herrera González:

* Clases/Modelos
* Componentes Nuevo

Conjunto:

* Componentes Update
* Componentes Consultar

**Agenda: (Objetivos)**

|  |  |
| --- | --- |
| Fecha: | Tareas |
| 21/08/09 | Creación del proyecto. Planteamiento del desarrollo del proyecto. Creación de clases/modelos y servicios CRUD |
| 21/08/10 | Componentes Login. Item/consultar. |
| 21/08/11 | Componente nav. Componentes para función nuevo. \*Comenzar a implementar CSS |
| 21/08/12 | Terminar updates. Terminar nav. Funciones pedido. Seguir con implementación CSS. Implementar Funcionalidades y transmisión de parámetros en Nav. |
| 21/08/13 | Terminar Funcionalidades pedido. Mensajes confirmación eliminar. Revisión funciones delete. |
| 21/08/14-15 | Corrección inserción/actualización de datos en BD con funcionalidades Pedido. Terminar de implementar CSS.  Revisión y corrección de errores. |

**Resultados diarios**

**09/08/21:**

* Creación del proyecto.
* Planteamiento del desarrollo del proyecto.
* Creación de clases/modelos y servicios CRUD.

**10/08/21:**

* Componente dashboard.
* Login.
* Componente usuario consultar.
* Componente artículo consultar.
* Componente pedido consultar.
* Funcionalidad delete.

**11/08/21:**

* Restructuración proyecto (Login, nav, dashboard).
* Componente usuario nuevo y update.
* Componente artículo nuevo y update.

**12/08/21:**

* Componente pedido consultar.
* Componente pedido nuevo.
* Implementación parcial CSS

**13/08/21:**

* Finalización componente pedido nuevo.
* Componente pedido update
* Revisión funcionalidades

**14-15/08/21:**

* Revisión funcionalidades
* Revisar/terminar componentes pedido
* Implementación restante de CSS
  1. ***Investigación***

Genericidad servicios Angular.

Uso de cookies.

Formularios reactivos. Validator.

Eventos JavaScript

* 1. ***Objetivo y Resultado***

El objetivo del proyecto es el desarrollo de una aplicación web con Angular 11. Las funcionalidades propuestas, las cuales se han implementado en su totalidad, son:

* Usuarios: Login, crear nuevo usuario, update usuario, eliminar usuario, consultar usuarios, búsqueda dinámica por nombre parcial.
* Artículos: Crear nuevo artículo, update artículo, eliminar artículo, consultar artículo, búsqueda dinámica por nombre.
* Pedidos: Crear nuevo pedido, update pedido, eliminar pedido, consultar pedido, búsqueda dinámica por nombre, actualizar stock de artículos con la creación, update y eliminación de los pedidos.
* Implementación de servicios a base de datos.

Se han encontrados dificultades en la interacción con la base de datos para actualizar el stock de artículos, a la hora de crear, modificar o eliminar un pedido.

* 1. ***Conclusión***

Se puede concluir que se ha cumplido el objetivo del proyecto obteniendo una aplicación Marketplace funcional.

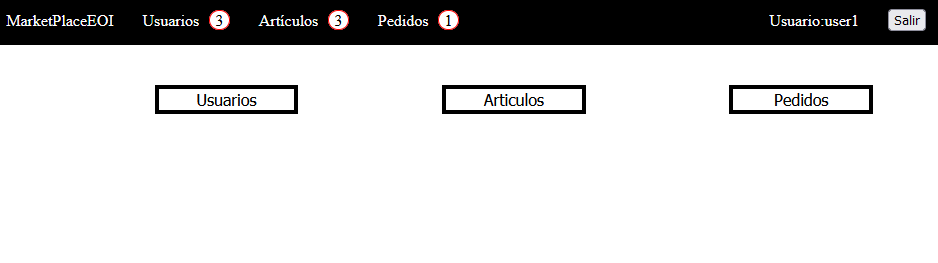
Ha sido un proyecto interesante en el que se han puesto en práctica los conocimientos aprendidos en el curso e incluso se han ampliado a partir de la información investigada. Con este proyecto se han podido asentar los conocimientos y ver que partes no habían quedado claras.

1. **Memoria Técnica**
   1. ***Pantallas Angular***

**Login:**

****

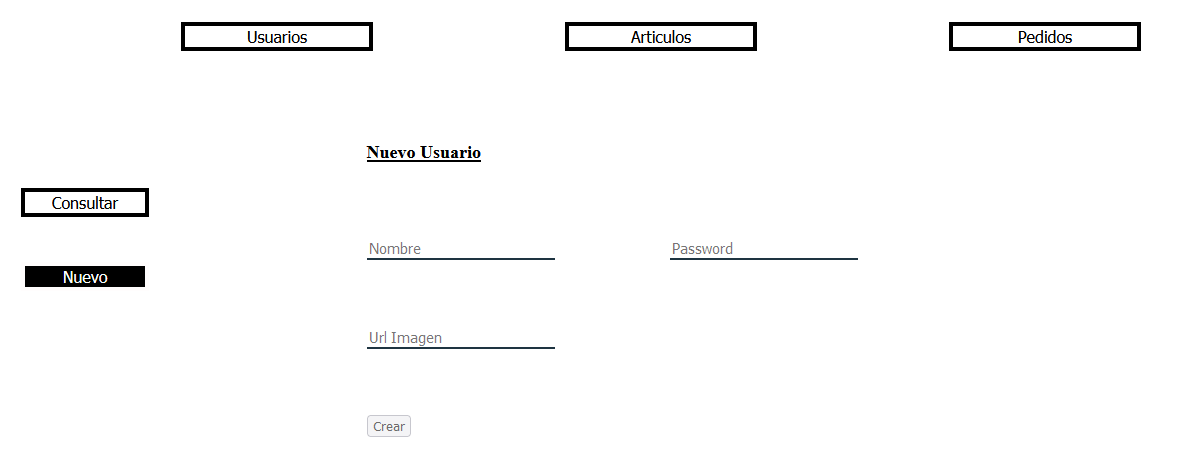
**Dashboard:**

****

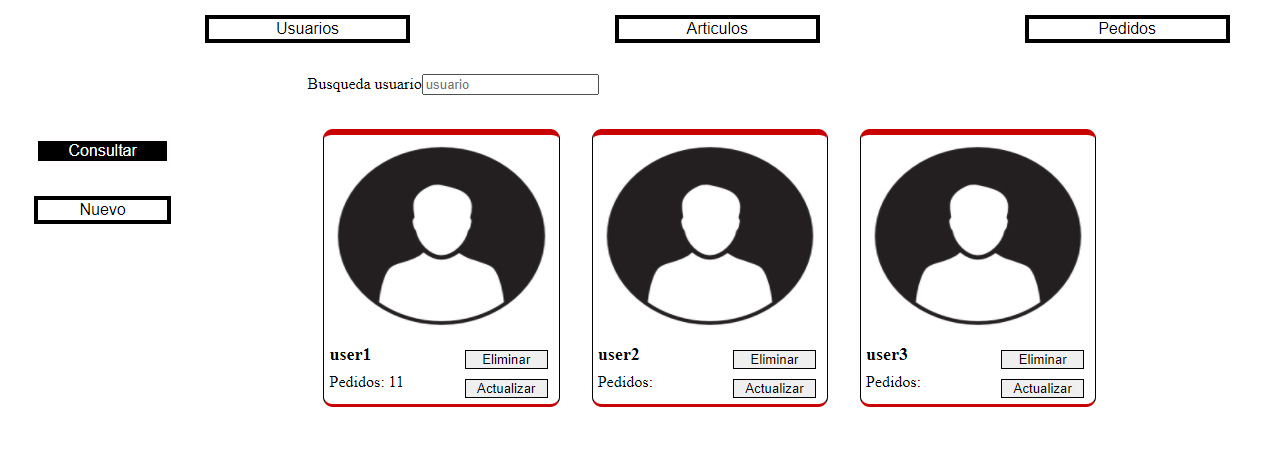
**Usuarios:**

****

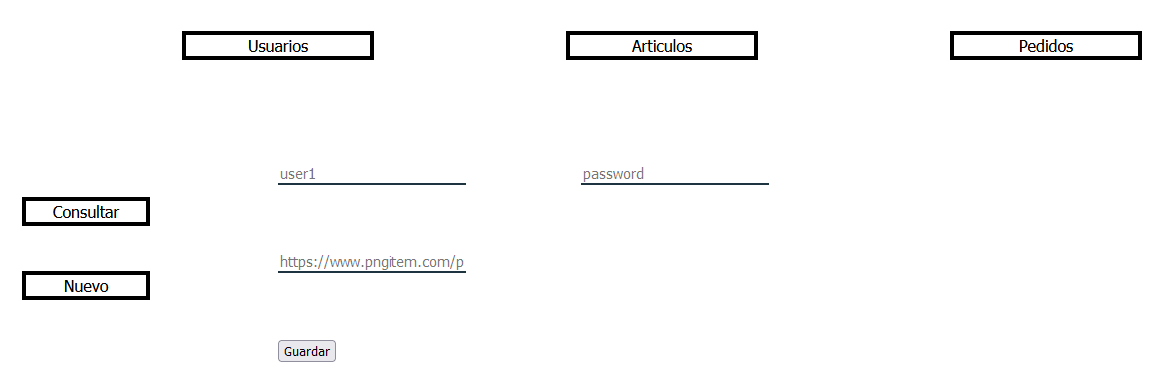
**Usuarios/Nuevo:**

****

**Usuarios/Consultar:**

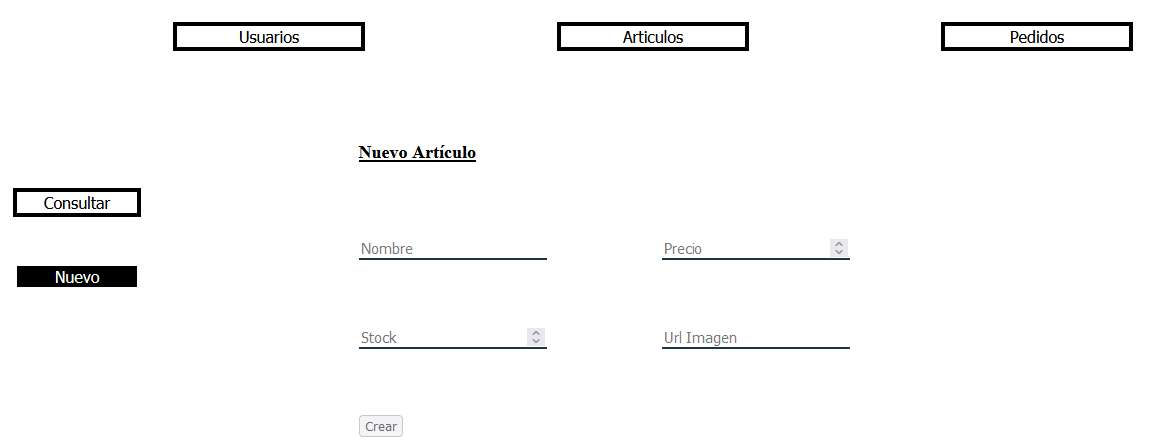
****

**Usuarios/Consultar/Update:**

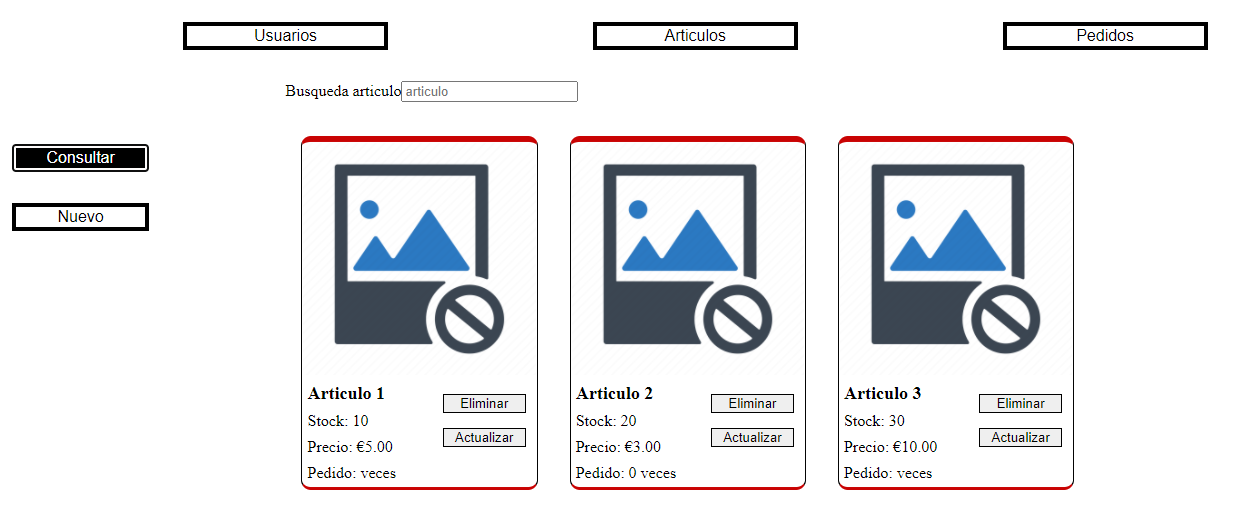
****

**Artículos:**

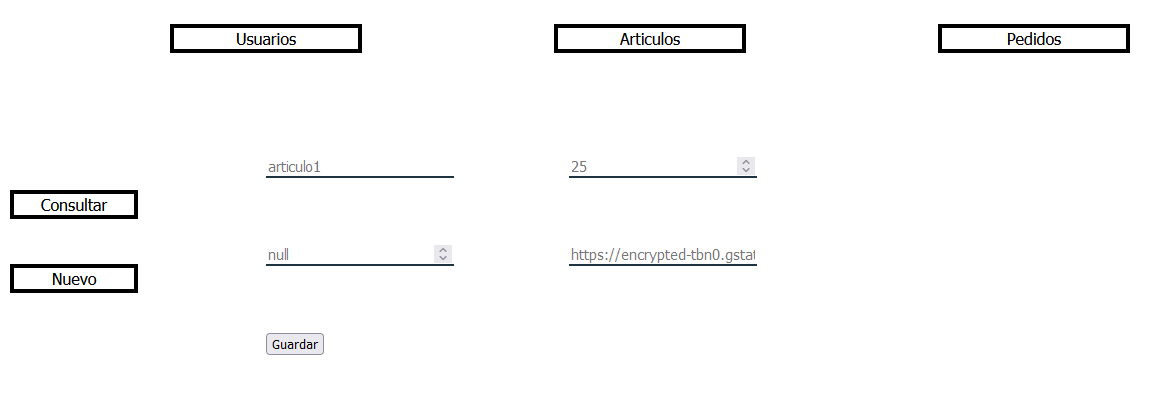
****

**Artículos/Nuevo:**

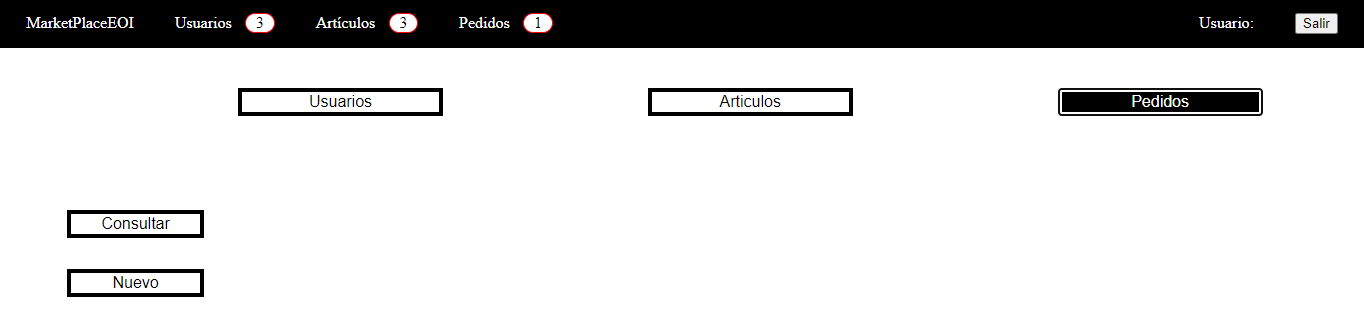
**Artículos/Consultar:**

****

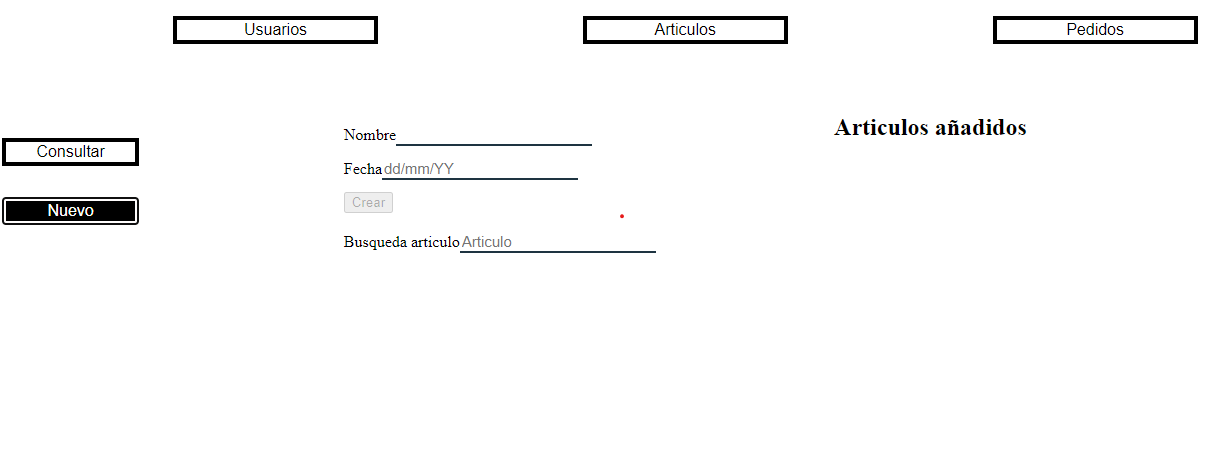
**Artículos/Consultar/Update:**

****

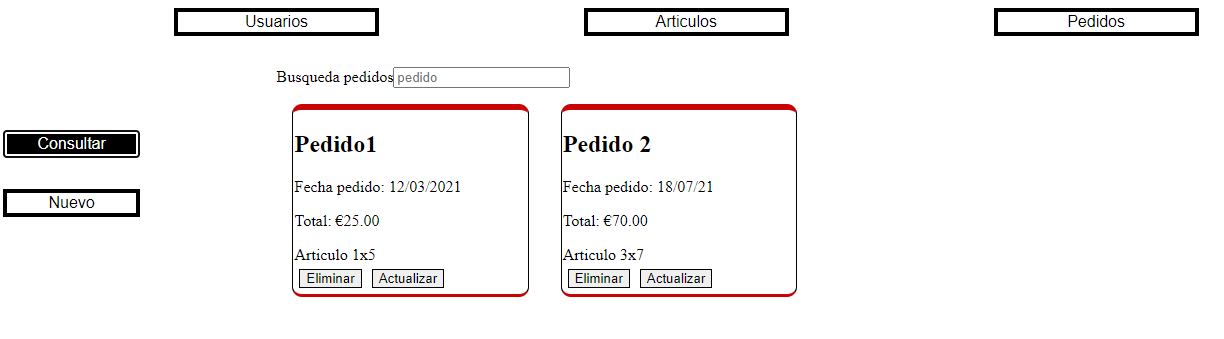
**Pedidos:**

****

**Pedidos/Nuevo:**

****

**Pedidos/Consultar:**

****

**Pedidos/Consultar/Update:**

****

* 1. ***Base de Datos***

*Json para JSonServer*



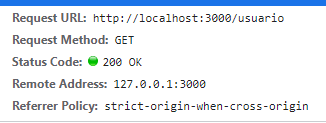
* 1. ***API REST***
     1. Servicio Web.

**Usuarios**

**Obtener Usuarios**

Obtención de todos los usuarios alojados en la base de datos utilizada.

**Entrada**:



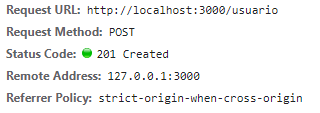
**Salida**:



**Crear Usuario**

Inserción de elemento en base de datos pasando en cuerpo de mensaje los datos del elemento utilizando método POST. Se hace una comprobación para no utilizar un nombre de usuario repetido. Si no se ha indicado ninguna URL de imagen se establece una por defecto.

**Entrada**:



**Salida**:

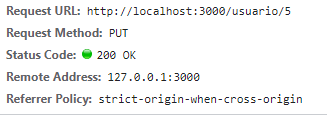


**Update Usuario**

Actualización de elemento alojado en base de datos pasando en cuerpo de mensaje los datos del elemento y el id como path variable. Se hace una comprobación para no utilizar un nombre de usuario repetido.

Si no se rellena alguno de los inputs del formulario, se dejarán con el valor que tenían anteriormente.

**Entrada**:



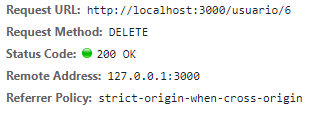
**Salida**:



**Delete Usuario**

Eliminación de usuario alojado en base de datos enviando su id.

**Entrada**:



**Salida**:

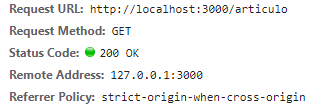


**Artículos**

**Obtener Artículos**

Obtención de todos los artículos alojados en la base de datos utilizada.

**Entrada**:



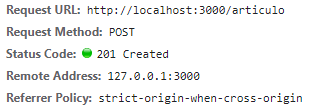
**Salida**:



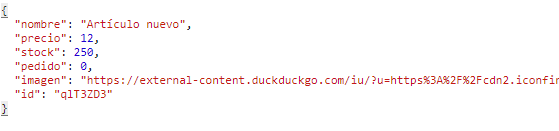
**Crear Articulo**

Inserción de elemento en base de datos pasando en cuerpo de mensaje los datos del elemento utilizando método POST. Se hace una comprobación para no utilizar un nombre de artículo repetido. Si no se ha indicado ninguna URL de imagen se establece una por defecto.

**Entrada**:



**Salida**:

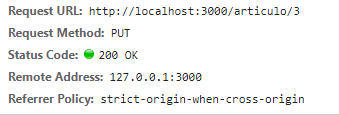


**Update Artículo**

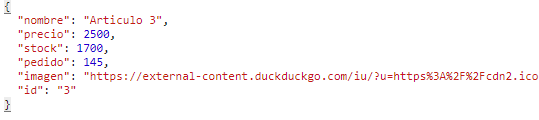
Actualización de elemento alojado en base de datos pasando en cuerpo de mensaje los datos del elemento y el id como path variable. Se hace una comprobación para no utilizar un nombre de artículo repetido.

Si no se rellena alguno de los inputs del formulario, se dejarán con el valor que tenían anteriormente.

**Entrada**:



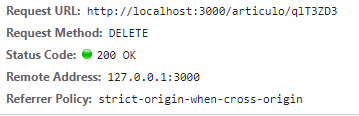
**Salida**:



**Delete Artículo**

Eliminación de artículo alojado en base de datos enviando su id.

**Entrada**:



**Salida**:

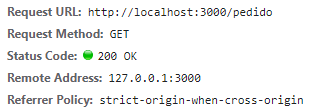


**Pedidos**

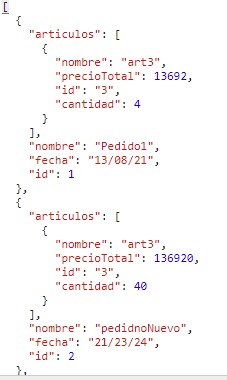
**Obtener Pedidos**

Obtención de todos los pedidos alojados en la base de datos utilizada.

**Entrada**:



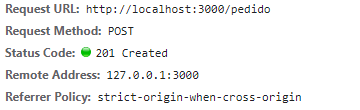
**Salida**:



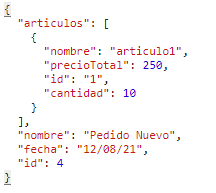
**Crear Pedido**

Inserción de elemento en base de datos pasando en cuerpo de mensaje los datos del elemento utilizando método POST. Se hace una comprobación para no utilizar un artículo repetido. Actualización de stock de productos.

**Entrada**:



**Salida**:

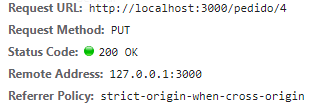
**

**Update Pedido**

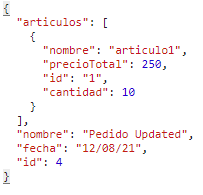
Actualización de elemento alojado en base de datos pasando en cuerpo de mensaje los datos del elemento y el id como path variable. Se hace una comprobación para no utilizar un artículo repetido.

Si no se rellena alguno de los inputs del formulario, se dejarán con el valor que tenían anteriormente.

**Entrada**:



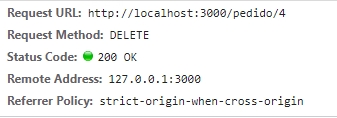
**Salida**:

**

**Delete Pedido**

Eliminación de pedido alojado en base de datos enviando su id.

**Entrada**:



**Salida**:

**