



FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS

MICROSERVICIOS

PROYECTO MARKETPLACE

PROYECTO ECOTU

GRUPO DIMEDAZO

**JUAN ALEJANDRO DIAZ LOTE
JUAN DAVID ALDANA RINCON
MARIANA ROJAS GALAVÍS
EDWIN FABIAN VESGA
SANTIAGO CAMACHO VILLEGAS**

27 DE MARZO 2023

1. Requisitos (CU o Historias de Usuario)

Cliente

- Como cliente quiero poder crear una cuenta con información como mi nombre, edad, usuario y contraseña para acceder al Marketplace.
- Como cliente quiero poder iniciar sesión para gestionar un carrito de compras y mi perfil.
- Como cliente quiero ver los diferentes servicios ofrecidos por la plataforma web.
- Como cliente quiero ver el detalle de un servicio en específico de la plataforma.
- Como cliente quiero poder buscar servicios ofrecidos por los proveedores por una palabra clave.
- Como cliente quiero poder calificar, comentar y hacer preguntas sobre un servicio en específico.
- Como cliente quiero utilizar el medio de pago registrado para hacer efectiva la adquisición de uno o varios servicios registrados en el carrito de compras.
- Como cliente quiero conocer la ubicación de los servicios para formar la ruta de llegada.
- Como cliente quiero conocer la información del clima durante la duración de mi viaje.

Proveedor

- Como proveedor quiero poder crear una cuenta con la información sobre la empresa o persona para publicar los servicios en el Marketplace.
- Como proveedor quiero poder iniciar mi cuenta para gestionar los servicios que ofrezco y mi perfil.
- Como proveedor quiero ver la calificación y comentarios que hacen de mis servicios.
- Como proveedor quiero poder dar respuesta a las preguntas que hacen sobre mis servicios.

2. Diagrama de clases

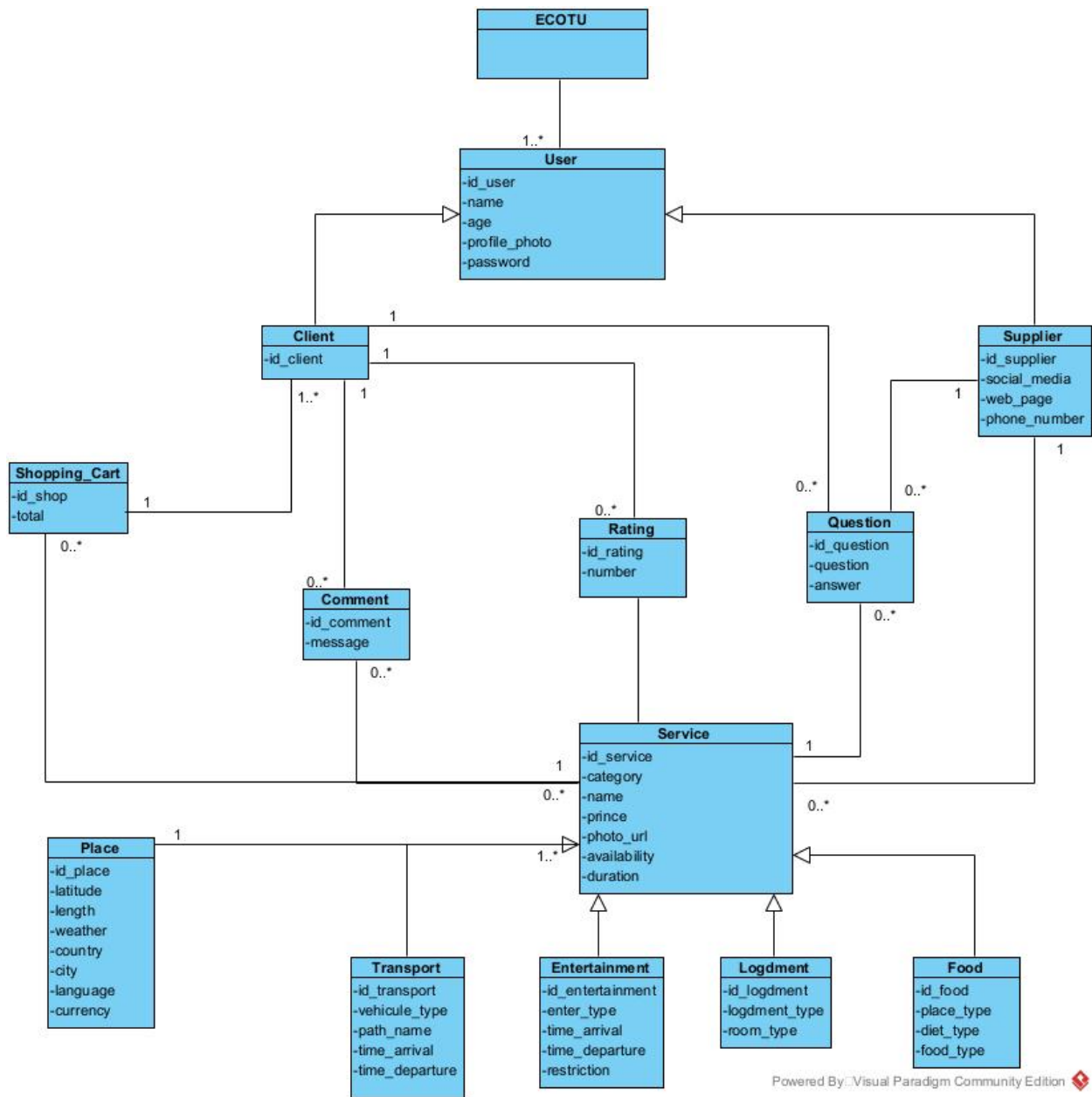


Ilustración 1. Diagrama Clases EcoTu

En el diagrama de clases anterior podemos ver la estructura de la aplicación propuesta para EcoTu.

Contamos con 14 clases que representan identidades del mundo real:

- **Usuario:** representa el usuario que tiene la aplicación EcoTu.
- **Cliente:** Tipo de usuario que representa a los interesados en adquirir un servicio en el Marketplace.
- **Proveedor:** Tipo de usuario que representa a las personas o empresas que quieren publicar los diferentes servicios/productos.

- Carrito de compras: Representa el lugar donde se van a encontrar los servicios que los usuarios quieren adquirir.
- Lugar: Representa el lugar donde está ubicado el servicio.
- Comentario: Representa los comentarios que puede tener cada servicio de la plataforma dejados por un usuario.
- Calificación: Representa la calificación que puede tener un servicio de la plataforma dejados por un usuario a cada servicio.
- Pregunta: Representa la pregunta que tiene un usuario sobre un servicio que puede ser contestada por el proveedor de este.
- Servicios: Representa el servicio que ofrece cada proveedor.
- Transporte: Tipo de servicio ofrecido .
- Alojamiento: Tipo de servicio ofrecido para el hospedaje de las personas.
- Alimentación: Tipo de servicio ofrecido .
- Entretenimiento: Tipo de servicio ofrecido referente a los planes como caminatas.

3. Persistencia (Diagrama lógico de datos o JSON Schema)

Para asegurar que durante el intercambio de información el lineamiento de los datos si se esté cumpliendo se va a utilizar JSON Schemas, con esto nosotros podemos garantizar que tanto los datos enviados como los recibidos si estén cumpliendo con nuestros parámetros.

Algunos ejemplos son los siguientes:

```

1 {
2   "$schema": "http://json-schema.org/draft-07/schema#",
3   "title": "CarritoCompras",
4   "type": "object",
5   "properties": {
6     "id_shop": {
7       "type": "string",
8       "description": "Identificador del carrito de compras",
9       "pattern": "^[a-f0-9]{24}$"
10    },
11    "total": {
12      "type": "number",
13      "description": "Total de la compra"
14    },
15    "id_client": {
16      "type": "string",
17      "description": "Identificador del cliente",
18      "pattern": "^[a-f0-9]{24}$"
19    },
20    "pay": {
21      "type": "boolean",
22      "description": "Indica si se realizó el pago"
23    }
24  },
25  "required": ["id_shop", "total", "id_client", "pay"]
26 }
27
1 {
2   "$schema": "http://json-schema.org/draft-07/schema#",
3   "title": "Alojamiento",
4   "type": "object",
5   "properties": {
6     "id_logdment": {
7       "type": "string",
8       "description": "Identificador del alojamiento",
9       "pattern": "^[a-f0-9]{24}$"
10    },
11    "logdment_type": {
12      "type": "string",
13      "description": "Tipo de alojamiento"
14    },
15    "room_type": {
16      "type": "string",
17      "description": "Tipo de habitación"
18    },
19    "id_service": {
20      "type": "string",
21      "description": "Identificador del servicio asociado al alojamiento",
22      "pattern": "^[a-f0-9]{24}$"
23    }
24  },
25  "required": ["id_logdment", "logdment_type", "room_type", "id_service"]
26 }
27

```

Figure 1. Ejemplo JSON Schemas

```

1  {
2    "$schema": "http://json-schema.org/draft-07/schema#",
3    "title": "cliente",
4    "type": "object",
5    "properties": {
6      "id_client": {
7        "type": "string",
8        "description": "Identificador del cliente",
9        "pattern": "^[a-f0-9]{24}$"
10     },
11     "username": {
12       "type": "string",
13       "description": "Nombre de usuario del cliente",
14       "minLength": 1,
15       "maxLength": 50
16     },
17     "name": {
18       "type": "string",
19       "description": "Nombre del cliente",
20       "minLength": 1,
21       "maxLength": 50
22     },
23     "age": {
24       "type": "integer",
25       "description": "Edad del cliente",
26       "minimum": 1,
27       "maximum": 150
28     },
29     "profile_photo": {
30       "type": "string",
31       "description": "URL de la foto de perfil del cliente",
32       "format": "uri"
33     },
34     "description": {
35       "type": "string",
36       "description": "Descripción del cliente",
37       "maxLength": 250
38     },
39   },
40 }

```

Figure 2. Ejemplo Json Schema

En este caso, el JSON Schema se va a utilizar para validar los datos que se reciben del usuario antes de enviarlos al servidor y viceversa. Se realizó un JSON schema para cada una de las entidades que se guardan en la base de datos, esto para poder garantizar que durante el intercambio de información o modificación de la misma la estructura de datos se mantenga igual.

4. Diseño de componentes (API REST)

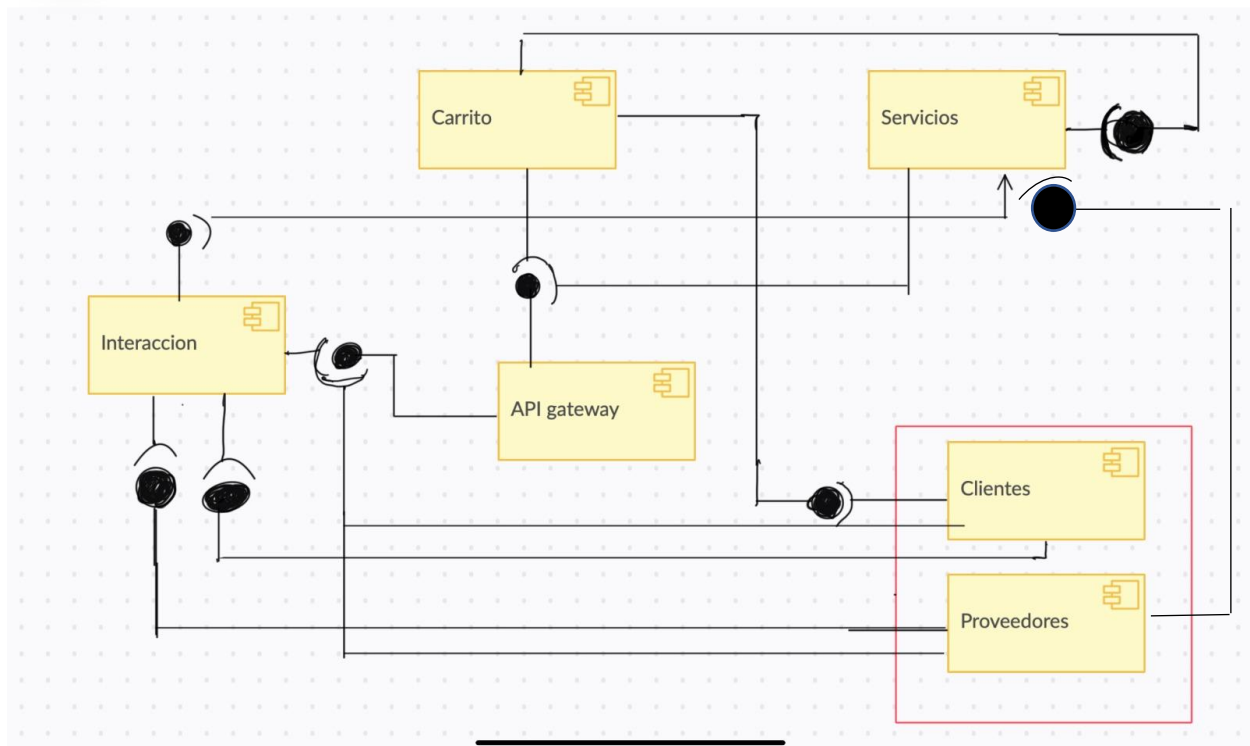
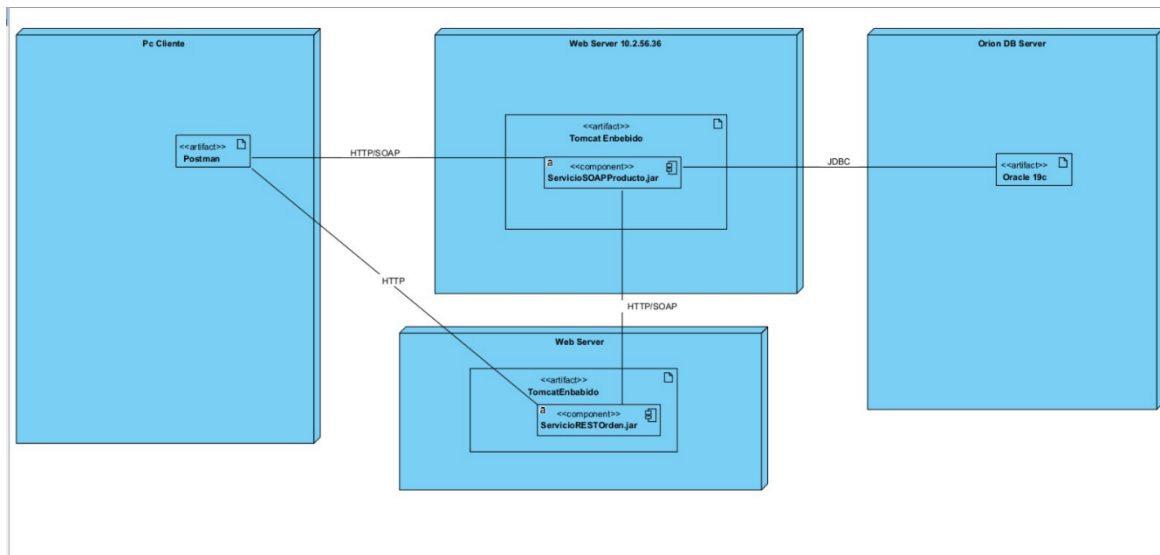


Figure 3. Diagrama componentes

En el diagrama de componentes mostrado vemos como todos los componentes ofrecen una interfaz a la apigateway para que este pueda gestionar sus diferentes interfaces, actuando como puerto de enlace para estos entre los que consumen y ofrecen. Además, tenemos el microservicio de carrito de compra el cual consume recibe datos de cliente y servicios para su correcto funcionamiento en el Marketplace. Para continuar tenemos el microservicio de interacción el cual ofrece a los clientes los servicios para que generen su pregunta sobre un producto y a los proveedores para que la respondan. Finalmente, el mcroservicios servicios por su parte ofrecen una interfaz para que los proveedores puedan gestionar sus productos en el Marketplace

Respecto a los servicios

5. Diagrama de Despliegue



Los dos componentes principales del diagrama son: Servicio Orden y Servicio Producto. La comunicación entre ambos servicios se realiza en HTTP/SOAP, por otro lado, para poder comunicarse con el cliente el servicio orden operado en REST se comunica con el cliente utilizando HTTP mientras que el Servicio Producto se comunica con el cliente utilizando HTTP/SOAP, este mismo mantiene una comunicación con el servidor de Orion, para esto utiliza JDBC. Cada uno de estos opera en nodos diferentes como se puede ver en el diagrama.