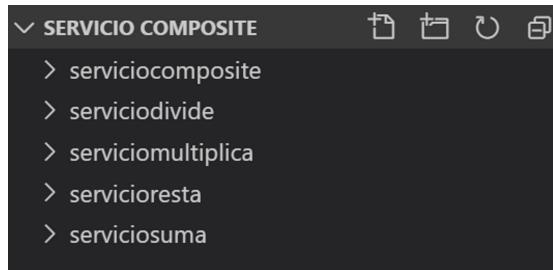


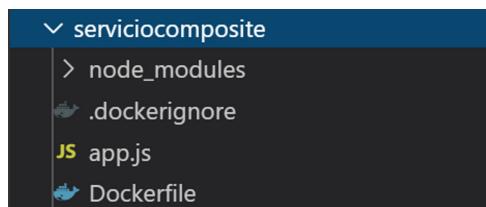
Composite

martes, 22 de octubre de 2019 12:19

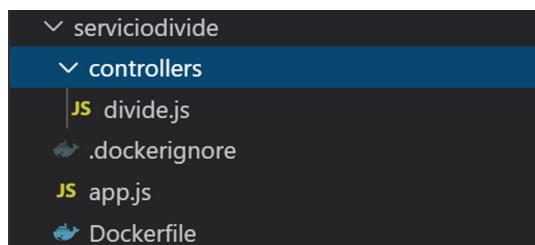
1. Crear estructura de servicios, una carpeta por cada servicio, 5 en total



2. Servicio composite (servicio de Tarea) orquesta el llamado de los otros containers (servidores) o microservicios



3. Estructura para cada servicio independiente, cada Microservicio tiene un controller (solo uno: divide.js, suma.js, resta.js, multiplica.js) y un api expuesta (app.js), 4 Microservicios



- 3.1 Controller de cada servicio, con la funcionalidad, en este caso no hay un model (abstracción de la BD)

A screenshot of a code editor showing the content of "divide.js". The code defines a division function that takes two parameters, num1 and num2, and returns their quotient. It includes console.log statements for debugging.

```
JS divide.js  ×
1 //Division function
2 exports.DivisionInq = function (req, res) {
3     var num1 = parseInt(req.params.num1);
4     var num2 = parseInt(req.params.num2);
5     var result;
6     console.log(num1);
7     console.log(num2);
8     if (num1 || num2) {
```

- 3.2 Cada servicio tiene un endpoint expuesto

```
JS app.js ×  
25     app.get('/mathOperations/division/:num1/:num2', CalController.DivisionInq);  
26
```

- 3.3 En el caso de composite las URLs orquestadas en app.js contendran el nombre del los servidores y el puerto (esta informacion está en el docker-compose.yml)

```
case "a":  
    request('http://addition:1000/mathOperations/addition/' + num1 + '/' + num2,  
    function (err, body) {  
        if (err) { return console.log(err); }  
  
case "s":  
    request('http://subtraction:2000/mathOperations/subtraction/' + num1 + '/' +  
    num2, function (err, body) {  
        if (err) { return console.log(err); }  
  
case "m":  
    request('http://multiplication:3000/mathOperations/multiplication/' + num1 + '/'  
    + num2, function (err, body) {  
        if (err) { return console.log(err); }  
  
case "d":  
    request('http://division:4000/mathOperations/division/' + num1 + '/' + num2,  
    function (err, body) {
```

- 3.4 Ambientar librerias y generar package.json con el init en cada carpeta (ojo) para cada uno de los servicios (5 servicios)

```
PS C:\Proyectos\Servicio Composite\serviciodivide> npm install --save express body-parser  
[...]  
[ 1.0.0] | fetchMetadata: sill resolveWithNewModule unpipe@1.0.0 checking installat
```

4. Generar el dockerfile en cada servicio (5 servicios)

```
Dockerfile ×  
1  FROM node:9-slim  
2  RUN mkdir /src  
3  WORKDIR /src  
4  COPY package*.json ./  
5  RUN npm install  
6  COPY . .  
7  CMD ["npm", "start"]
```

- 4.1 crear el .dockerignore para subir solo lo necesario a la imagen (5 servicios)

```
↳ .dockerignore ×  
1 node_modules  
2 .git/
```

4.2 npm init (en cada carpeta de cada servicio) (5 servicios)

```
✓ serviciodivide  
  ✓ controllers  
    JS divide.js  
  > node_modules  
  ↳ .dockerignore  
  JS app.js  
  Dockerfile  
  {} package-lock.json  
  {} package.json
```

```
{} package.json ×  
1 {  
2   "name": "ms-division",  
3   "version": "1.0.0",  
4   "description": "servicio de division 2 parametros requeridos",  
5   "main": "app.js",  
6   "dependencies": {  
7     "body-parser": "^1.19.0",  
8     "express": "^4.17.1"  
9   },  
10 }
```

4.3 Editar el package.json agregando el script de arranque, la linea:
"start": "node app.js",

```
},  
"devDependencies": {},  
"scripts": {  
  "start": "node app.js",  
  "test": "echo \\\"Error: no test specified\\\" && exit 1"  
},  
"keywords": [  
  "divide"  
]
```

5. Correr el aplicativo, corregir errores y probar en Postman que la URL funcione con los parametros solo para los 4 servicios

5.1 Para probar el composite, debera cambiar nombres de los servidores por localhost y asignarle un puerto diferente a cada microservicio

```
case "a":  
  request('http://localhost:1000/mathOperations/addition/' + num1 + '/' + num2,  
  function (err, body) {  
    if (err) { return console.log(err); }  
  })  
  .on('data', function (data) {  
    console.log(data);  
  })  
  .end();  
}  
});
```

```

case "s":
    request('http://localhost:2000/mathOperations/subtraction/' + num1 + '/' +
    num2, function (err, body) {
        if (err) { return console.log(err); }
        result = JSON.parse(body);
    });

case "m":
    request('http://localhost:3000/mathOperations/multiplication/' + num1 + '/' +
    num2, function (err, body) {
        if (err) { return console.log(err); }
    });

case "d":
    request('http://localhost:4000/mathOperations/division/' + num1 + '/' + num2,
    function (err, body) {
        if (err) { return console.log(err); }
    });

```

6. Construir la imagen de cada servicio y validar su creación

----Crear imagen de la division

`docker build -t ms-division .`

----Crear imagen de la multiplicacion

`docker build -t ms-multiplication .`

----Crear imagen de la resta

`docker build -t ms-subtraction .`

----Crear imagen de la suma

`docker build -t ms-addition .`

----Crear el imagen del orquestador

`docker build -t ms-composite .`

```

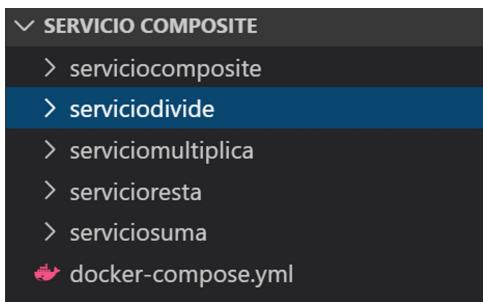
PS C:\Proyectos\Servicio-Composite\serviciodivide> docker build -t ms-division .
Sending build context to Docker daemon 99.33kB
Step 1/7 : FROM node:9-slim
--> e20bb4abe4ee
Step 2/7 : RUN mkdir /src
--> Using cache
--> 56900ff3b150

```

docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ms-multiplication	latest	86285efa44ab	11 seconds ago	185MB
ms-division	latest	5b03b5ffd988	51 seconds ago	185MB
ms-subtraction	latest	f8ea7bc4cbb5	About a minute ago	185MB
ms-addition	latest	6ea017d92df3	2 minutes ago	185MB
ms-composite	latest	d1705e0594bb	2 minutes ago	189MB

7. Crear el docker-compose.yml donde se organizan los contenedores, se crea fuera de los 5 servicios a nivel del proyecto



```

version: '3'
#Declarar los servicios
#Cada servicio con la imagen ya creada
#expose para comunicacion interna (este puerto debe ser el mismo que se declaro en cada api llamada dentro del app del composite) y ports para exponer al exterior
#depends_on para ligar conexion entre contenedores
#Crear la red domain.calculus antes de ligarlos
#networks para agrupar los contenedores y solo se comuniquen entre esa red
services:
  addition:
    container_name: addition
    image: ms-addition
    expose:
      - '1000'
    networks:
      - domain.calculus
  subtraction:
    container_name: subtraction
    image: ms-subtraction
    expose:
      - '2000'
    networks:
      - domain.calculus
  multiplication:
    container_name: multiplication
    image: ms-multiplication
    expose:
      - '3000'
    networks:
      - domain.calculus
  division:
    container_name: division
    image: ms-division
    expose:
      - '4000'
    networks:
      - domain.calculus
  composite:
    container_name: composite
    image: ms-composite
    ports:
      - '5000:1000'
    networks:
      - domain.calculus
    depends_on:
      - multiplication
      - division
      - addition
      - subtraction
networks:
  domain.calculus:
    external: true

```

8. Crear la red y validar que se creo

docker network create domain.calculus

docker network ls

```
PS C:\Proyectos\Servicio-Composite\servicioresta> docker network create domain.calculus
f33592fbc9b362f4466b6a951a633439e24db498562042612ab833377bedc92d
PS C:\Proyectos\Servicio-Composite\servicioresta> docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
d1993753f3c7    bridge    bridge      local
f33592fbc9b3    domain.calculus    bridge      local
945c857e2057    host      host      local
ff7071c08575    none      null      local
```

9. Crear el paquete de servicios (contenedores), fuera de las carpetas de los servicios

Mode	LastWriteTime	Length	Name
---	-----	-----	-----
d----	28/10/2019	20:57	.vscode
d----	9/11/2019	10:48	serviciocomposite
d----	28/10/2019	20:56	serviciodivide
d----	28/10/2019	20:56	serviciomultiplica
d----	28/10/2019	20:55	servicioresta
d----	28/10/2019	12:47	serviciosuma
-a----	28/10/2019	09:53	1254 docker-compose.yml

docker-compose up -d

```
PS C:\Proyectos\Servicio-Composite> docker-compose up -d
Creating addition     ... done
Creating multiplication ... done
Creating subtraction   ... done
Creating division     ... done
Creating composite     ... done
```

9.1 | Validar la creacion de contenedores

docker ps

CONTAINER ID	IMAGE NAMES	COMMAND	CREATED	STATUS	PORTS
0f7737eed1d9	ms-composite	"npm start"	2 minutes ago	Up 2 minutes	0.0.0.0:
5000->1000/tcp	composite				
973b372d43ef	ms-addition	"npm start"	2 minutes ago	Up 2 minutes	1000/tcp
1d53c7c1cfef	addition				
e6366364b3c7	ms-multiplication	"npm start"	2 minutes ago	Up 2 minutes	1000/tcp
ac3655062f49	multiplication				
	ms-division	"npm start"	2 minutes ago	Up 2 minutes	1000/tcp
	division				
	ms-subtraction	"npm start"	2 minutes ago	Up 2 minutes	1000/tcp
	subtraction				

9.2 | Validar que los contenedores estan en red, deben estar los 5 servicios

docker network inspect domain.calculus

```
"Containers": {
    "38f951fd81f074f731e074f7b3382c1767e5b2a2
        "Name": "subtraction",
        "EndpointID": "d0fb29b85165434faa8260
        "MacAddress": "02:42:ac:16:00:04",
        "IPv4Address": "172.22.0.4/16",
        "IPv6Address": ""
    },
    "68fbf88164c7bc0e2a052ea25b9608eaa46d5b8a
        "Name": "multiplication",
        "EndpointID": "31e24784a5076dea48a667
    }

    "973b372d43ef16d5df7944f6a59e53a28eacaba82
        "Name": "addition",
        "EndpointID": "94abf07f93abde46d30151d
        "MacAddress": "02:42:ac:13:00:05",
        "IPv4Address": "172.19.0.5/16",
        "IPv6Address": ""

    },
    "bd40134cf562afaa77fd75f24fecac9d838
        "Name": "composite",
        "EndpointID": "8395346e064f0a941
        "MacAddress": "02:42:ac:16:00:06",
        "IPv4Address": "172.22.0.6/16",
        "IPv6Address": ""
    },
    "dddf185ae301a25bd965c5a93c4f90ad64f
        "Name": "division",
        "EndpointID": "959bc2b836cac4d2a
        "MacAddress": "02:42:ac:16:00:02
        "IPv4Address": "172.22.0.2/16"
    }
}
```

10. Probar el funcionamiento del servicio orquestador y los otros servicios mediante los logs de cada servicio

```
PS C:\Proyectos\Servicio-Composite> docker logs composite

> ms-composite@1.0.0 start /src
> node app.js

Server Inicializado en el puerto: 1000
PS C:\Proyectos\Servicio-Composite> docker logs addition

> ms-addition@1.0.0 start /src
> node app.js

Server addition Inicializado en el puerto: 1000
PS C:\Proyectos\Servicio-Composite> docker logs subtraction

> ms-subtraction@1.0.0 start /src
> node app.js

Server subtraction Inicializado en el puerto: 2000
```

```
PS C:\Proyectos\Servicio-Composite> docker logs multiplication

> ms-multiplication@1.0.0 start /src
> node app.js

Server multiplication inicializado en el puerto: 3000
PS C:\Proyectos\Servicio-Composite> docker logs division

> ms-division@1.0.0 start /src
> node app.js

Server division Inicializado en el puerto: 4000
```

GET ▼ localhost:5000/mathOperations/:operation/:num1:num2

operation	a
num1	10
num2	5

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize BETA JSON ▼

```
1  [
2   "Entradas": {
3     "Operacion": "a",
4     "Numero1": 10,
5     "Numero2": 5
6   },
7   "Salidas": {
8     "Resultado": "15"
9   }
10 ]
```

GET localhost:5000/mathOperations/:operation/:num1/:num2

operation	s
num1	10
num2	5

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1  [
2      "Entradas": {
3          "Operacion": "s",
4          "Numero1": 10,
5          "Numero2": 5
6      },
7      "Salidas": {
8          "Resultado": "5"
9      }
10 ]
```

GET localhost:5000/mathOperations... + ...

GET localhost:5000/mathOperations/:operation/:num1/:num2

operation	m
num1	10
num2	5

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1  [
2      "Entradas": {
3          "Operacion": "m",
4          "Numero1": 10,
5          "Numero2": 5
6      },
7      "Salidas": {
8          "Resultado": "50"
9      }
10 ]
```

GET localhost:5000/mathOperations/:operation/:num1/:num2

operation	d
num1	10
num2	5

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize BETA JSON

```
1  [
2   "Entradas": {
3     "Operacion": "d",
4     "Numero1": 10,
5     "Numero2": 5
6   },
7   "Salidas": {
8     "Resultado": "2"
9   }
10 ]
```

```
PS C:\Proyectos\Servicio-Composite> docker logs addition

> ms-addition@1.0.0 start /src
> node app.js

Server addition Inicializado en el puerto: 1000
-----
10
5
15
-----
```

```
PS C:\Proyectos\Servicio-Composite> docker logs subtraction

> ms-subtraction@1.0.0 start /src
> node app.js

Server subtraction Inicializado en el puerto: 2000
-----
10
5
5
-----
```

```
PS C:\Proyectos\Servicio-Composite> docker logs multiplication

> ms-multiplication@1.0.0 start /src
> node app.js

Server multiplication inicializado en el puerto: 3000
-----
10
5
50
-----
```

```
PS C:\Proyectos\Servicio-Composite> docker logs division

> ms-division@1.0.0 start /src
> node app.js

Server division Inicializado en el puerto: 4000
-----
10
5
2
-----
```

11. Validar que no se pueda accesar desde el puerto 1000,2000,3000,4000 a cada servicio según lo declarado en el app.js del composite ver punto 5.1

GET ▾ localhost:1000/mathOperations/addition/:num1/:num2

KEY	VALUE
num1	10
num2	5

Could not get any response

There was an error connecting to localhost:1000/mathOperations/addition/10/5.

GET ▾ localhost:2000/mathOperations/subtraction/:num1/:num2

KEY	VALUE
num1	10
num2	5

Could not get any response

There was an error connecting to localhost:2000/mathOperations/subtraction/10/5.

GET ▾ localhost:3000/mathOperations/multiplication/:num1/:num2

KEY	VALUE
num1	10
num2	5

Could not get any response

There was an error connecting to localhost:3000/mathOperations/multiplication/10/5.

GET localhost:4000/mathOperations/division/:num1:num2

KEY	VALUE
num1	10
num2	5

Could not get any response

There was an error connecting to localhost:4000/mathOperations/division/10/5.

12. Subir cada servicio al hub, mismo proceso para cada uno

12.1	Hacer commit al contenedor
12.2	Hacer push al contenedor para subirlo

```
PS C:\Proyectos\Servicio Composite> docker commit 68fbf88164c7 bandala/ms-multiplication:v1
sha256:9df324cc541c876a2774ad527b2a7a7a93a3587e952980fbe8bbc3d4200c8a0e
PS C:\Proyectos\Servicio Composite> docker push bandala/ms-multiplication:v1
The push refers to repository [docker.io/bandala/ms-multiplication]
128c9e14b6e0: Pushing [=====] 4.608kB
17e5aac74afb: Pushing [=====] 100.4kB
f892a6c12748: Pushing [=====] 8.822MB
```

```
PS C:\Proyectos\Servicio Composite> docker commit dddff185ae301 bandala/ms-division:v1
sha256:431837987611198ecbe8a3427f35085c325be293e8a26310f357cbd3e177d8ae
PS C:\Proyectos\Servicio Composite> docker push bandala/ms-division:v1
The push refers to repository [docker.io/bandala/ms-division]
3d34a5e7c0dc: Pushed
2f3ac5693bda: Pushed
218015552071: Pushed
```

```
PS C:\Proyectos\Servicio Composite> docker commit b0a94c5dd46a bandala/ms-addition:v1
sha256:3f8c5f7165be763dd9ee86533a0a462fd16b7bdcf00f358d92ac64ed10fa633a
PS C:\Proyectos\Servicio Composite> docker push bandala/ms-addition:v1
The push refers to repository [docker.io/bandala/ms-addition]
b0dcdf00428f: Pushed
```

```
PS C:\Proyectos\Servicio Composite> docker commit 38f951fd81f0 bandala/ms-subtraction:v1
sha256:662d3275e6f3731f298ae6364631efef337f48386f83e2c8054c9e579c484337
PS C:\Proyectos\Servicio Composite> docker push bandala/ms-subtraction:v1
The push refers to repository [docker.io/bandala/ms-subtraction]
e981de288e89: Pushed
```

```
PS C:\Proyectos\Servicio Composite> docker commit bd40134cf562 bandala/ms-composite:v1
sha256:456a86a47d5e21d54c519532567b07be5e7bad9ad76f1e95ccdb881c266919e9
PS C:\Proyectos\Servicio Composite> docker push bandala/ms-composite:v1
The push refers to repository [docker.io/bandala/ms-composite]
765a55511d2: Pushed
```