EJERCICIOS POSTMAN

POR: Morgan Aylen Posada Escobar

ACT 1: IMC ****

→ SCRIPTS:

```xml
<?xml version='1.0' encoding="UTF-8"?>
<methodCall>
    <methodName>Methods.getData</methodName>
    <params>
        <param>
            <value><double>59.0</double></value>
            <value><double>1.62</double></value>
        </param>
    </params>
</methodCall>
```
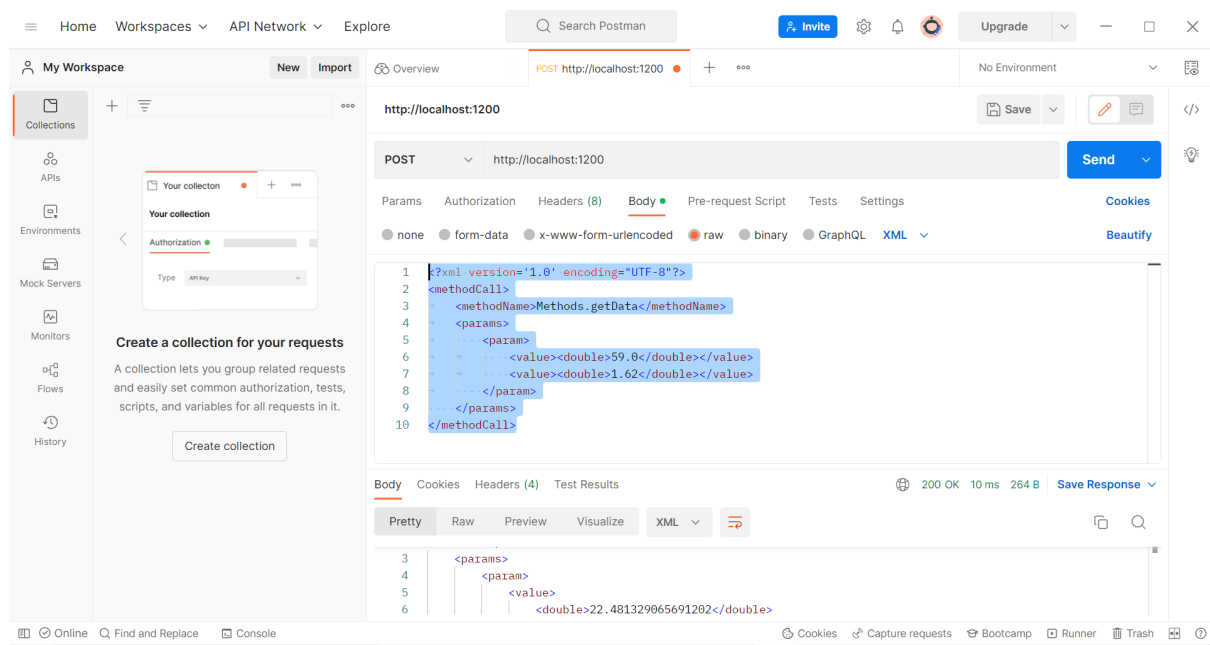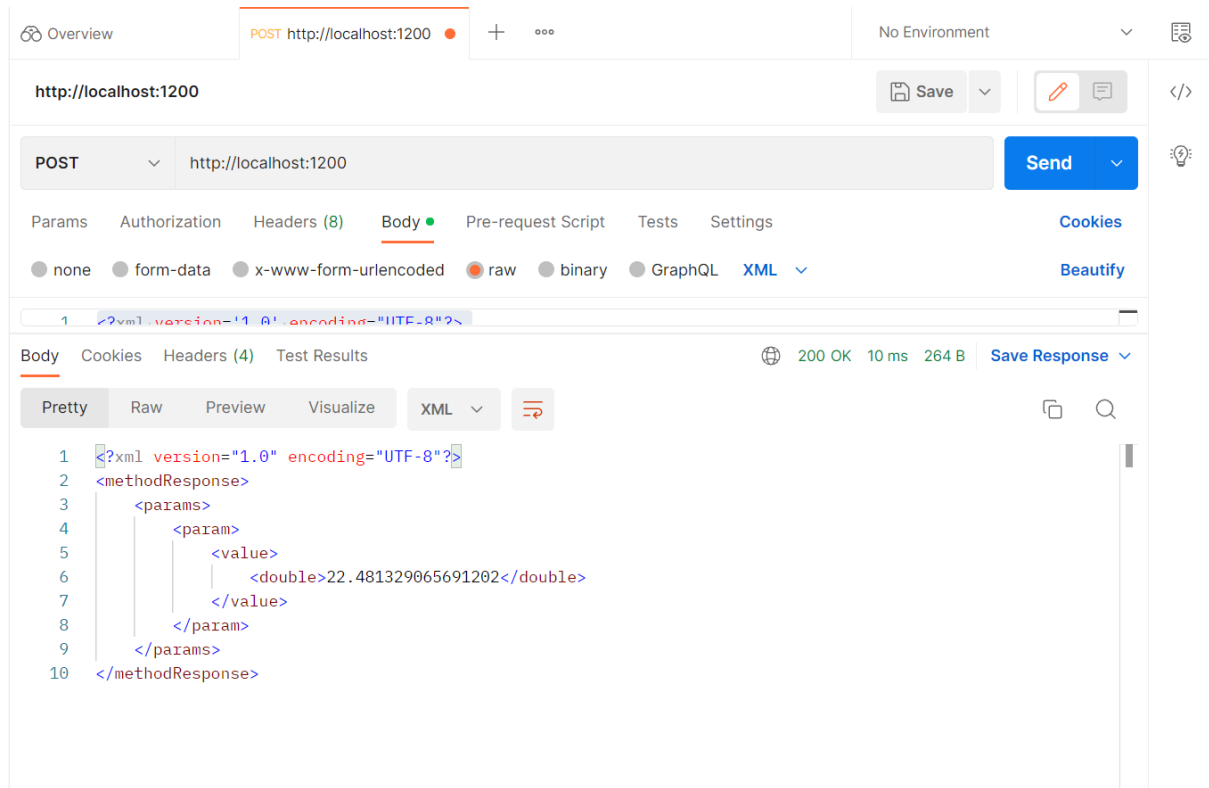
→ CAPTURAS:

https://abi.gitbook.io/net-core/4.-creando-tu-primer-servicio/4.4-probando-tus-servicios-con-postman

POR: Morgan Aylen Posada Escobar

ACT 2: Leer 4 variables y hacer operaciones; suma, producto y promedio

```xml
<?xml version='1.0' encoding="UTF-8"?>
<methodCall>
    <methodName>Methods.operations</methodName>
    <params>
        <param>
            <value><double>2</double></value>
            <value><double>3</double></value>
            <value><double>4</double></value>
            <value><double>6</double></value>
        </param>
    </params>
</methodCall>
```
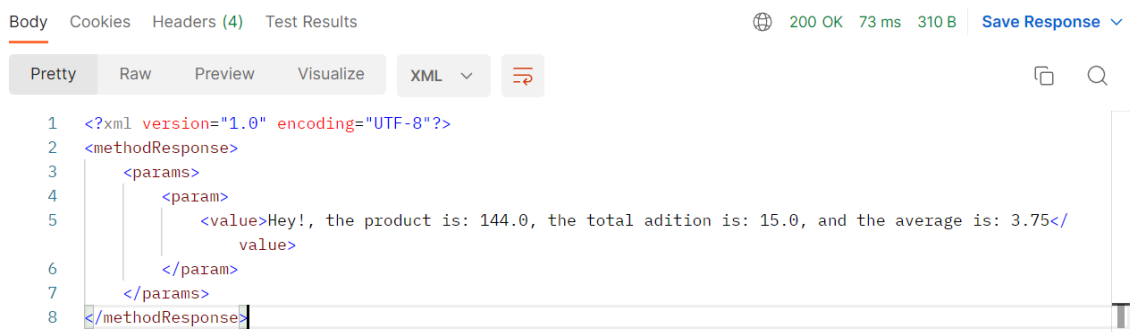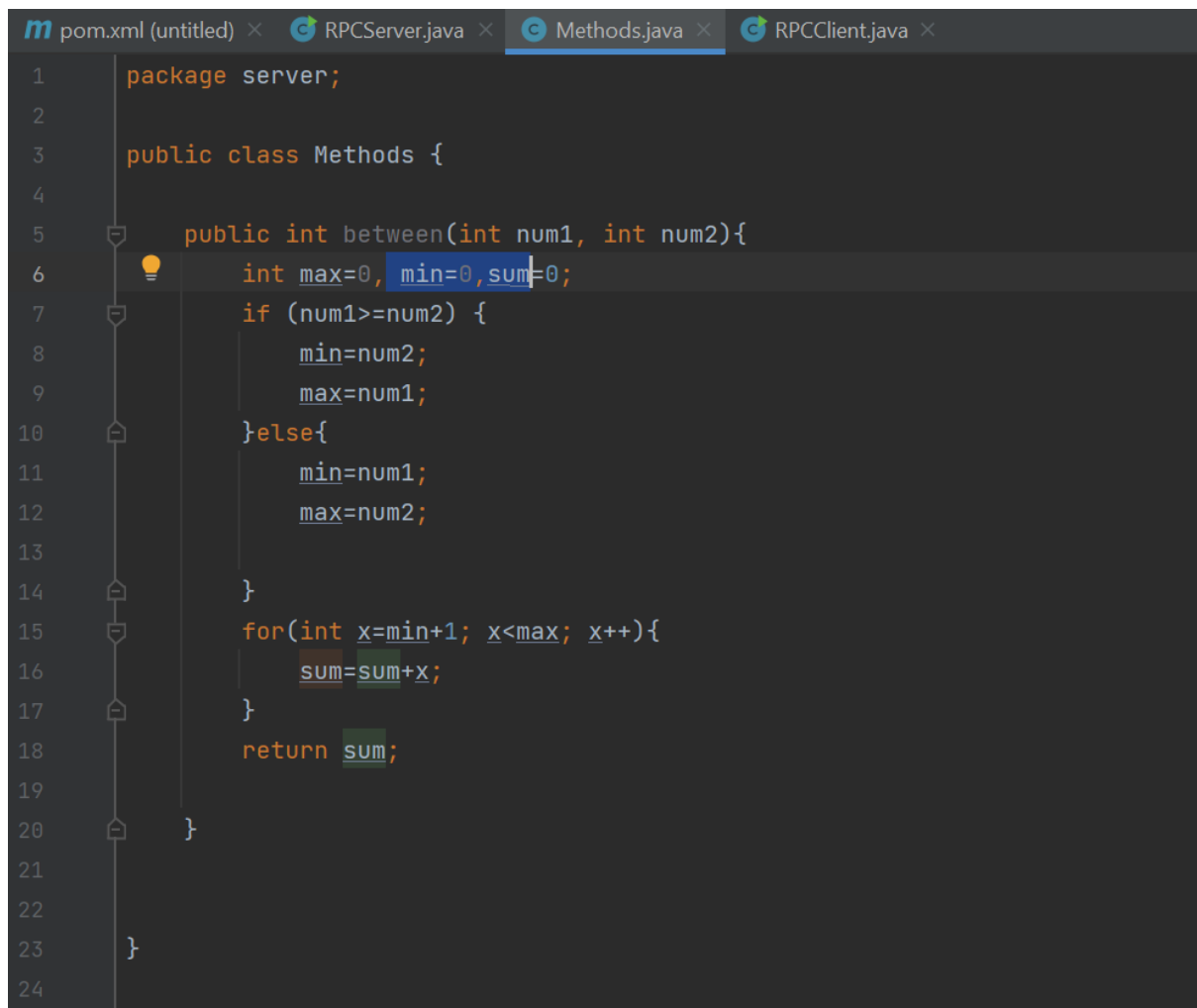
POR: Morgan Aylen Posada Escobar

| Body | Cookies | Headers (4) | Test Results | | 200 OK | 73 ms | 310 B | Save Response ⌄ |

| Pretty | Raw | Preview | Visualize | XML ⌄ |

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <methodResponse>
3      <params>
4          <param>
5              <value>Hey!, the product is: 144.0, the total adition is: 15.0, and the average is: 3.75</value>
6          </param>
7      </params>
8  </methodResponse>
```

ACT 3: Suma de los números que se encuentran entre dos números

```java
package server;

public class Methods {

    public int between(int num1, int num2){
        int max=0, min=0,sum=0;
        if (num1>=num2) {
            min=num2;
            max=num1;
        }else{
            min=num1;
            max=num2;

        }
        for(int x=min+1; x<max; x++){
            sum=sum+x;
        }
        return sum;

    }

}
```

```xml
<methodCall>
    <methodName>Methods.between</methodName>
    <params>
        <param>
            <value><int>10</int></value>
```

POR: Morgan Aylen Posada Escobar

```xml
        <value><int>15</int></value>
    </param>
  </params>
</methodCall>
```

Body   Cookies   Headers (4)   Test Results

Pretty   Raw   Preview   Visualize   XML ∨

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <methodResponse>
3      <params>
4          <param>
5              <value>
6                  <i4>50</i4>
7              </value>
8          </param>
9      </params>
10 </methodResponse>
```

ACT 4: arrays que ordene de forma ascendente

```java
package server;

import java.lang.reflect.Array;
import java.util.Arrays;

1 usage
public class Methods {

    public String ascendent(int num1, int num2, int num3, int num4, int num5){
        Object[] values={num1, num2, num3, num4, num5};
        Arrays.sort(values);
        return ("The number order is: "+Arrays.toString(values));
    }
}
```

```xml
<?xml version='1.0' encoding="UTF-8"?>
<methodCall>
```

POR: Morgan Aylen Posada Escobar

```xml
<methodName>Methods.ascendent</methodName>
<params>
   <param>
      <value><int>10</int></value>
      <value><int>15</int></value>
      <value><int>3</int></value>
      <value><int>4</int></value>
      <value><int>5</int></value>
   </param>
</params>
</methodCall>
```

Body   Cookies   Headers (4)   Test Results   ⊕   200 O

Pretty   Raw   Preview   Visualize   XML ∨   ⇄

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <methodResponse>
3      <params>
4          <param>
5              <value>The number order is: [3, 4, 5, 10, 15]</value>
6          </param>
7      </params>
8  </methodResponse>
```