

15^a
Emisión

DIPLOMADO Desarrollo de Sistemas con Tecnología Java

Módulo 5-6 Desarrollo de aplicaciones empresariales con Jakarta EE

Uriel Hernández



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC



Educación
Continua
1971 - 2021



SERVICIOS WEB RESTFUL

REST

- REST es el acrónimo para Representational State Transfer.
- Propuesto por el Dr. Roy Fielding en su tesis de doctorado.
- La arquitectura REST es un estándar de comunicación sin estado para aplicaciones modernas basado en HTTP.
- Es un estilo arquitectural para sistemas de hipermedia distribuidos.

Principios REST

- Interfaz Uniforme:
 - Identificación de Recursos: La interfaz debe identificar de forma única cada recurso
 - Manipulación de Recursos a través de representaciones: Los recursos deben tener representaciones uniformes en la respuesta del servidor.
- Cliente-Servidor: El patrón de diseño Cliente-Servidor promueve la separación de responsabilidades, lo que permite que los componentes evolucionen de forma independiente en tanto las interfaces/contratos no sean modificados.
- Stateless (Sin Estado): Establece que cada request del cliente al servidor debe contener toda la información necesaria para entender y completar el request.

Métodos HTTP

Acción	Path HTTP (Endpoint)	Retorno	Método HTTP	Código HTTP
Crear usuario	/usuarios	Usuario creado	POST	201 - CREATED 409 - CONFLICT
Actualizar usuario	/usuarios/{idUserio}	Usuario actualizado	PUT	200 - OK 204 - NO CONTENT 404 - NOT FOUND
Obtener usuario por Id	/usuarios/{idUserio}	Usuario o no encontrado	GET	200 - OK 404 - NOT FOUND
Obtener todos los usuarios	/usuarios	Listado de usuarios	GET	200 - OK
Eliminar usuario	/usuarios/{idUserio}	Void	DELETE	204 - NO CONTENT

Anotaciones JAX-RS

- **@ApplicationPath**: Define el recurso root de la aplicación, por lo que todos los recursos deben accederse a partir de esta base.
- **@Path**: Permite definir el path de las urls que atenderá el REST resource.
- **@Consumes**: Define los tipos MIME (Multipurpose Internet Mail Extensions) que los métodos de la clase recurso pueden consumir.
- **@Produces**: Define los tipos MIME (Multipurpose Internet Mail Extensions) que los métodos de la clase recurso pueden producir.

Anotaciones JAX-RS

- **@GET**: Método a invocar cuando se solicita obtener los datos de un recurso vía una petición HTTP GET.
- **@POST**: Método a invocar cuando se solicita crear un nuevo recurso vía una petición HTTP POST.
- **@PUT**: Método a invocar cuando se solicita actualizar un recurso existente vía una petición HTTP PUT.
- **@DELETE**: Método a invocar cuando se solicita eliminar un recurso existente vía una petición HTTP DELETE.

Anotaciones JAX-RS

- **@PathParam:** Permite ligar el parámetro especificado en el path e inyectarlo en el método java a ejecutarse.

Ejemplo: /path/usuarios/2788


- **@QueryParam:** Permite ligar el parámetro de consulta HTTP especificado en el path e inyectarlo en el método java a ejecutarse, muy similar a PathParam. Un parámetro de consulta HTTP se especifica por una clave valor después de un signo ?

Ejemplo: /path/usuarios?email=email@gmail.com

Postman: Workspace pixup

Create your workspace

Get the most out of your workspace with a template.


 Blank workspace 


Explore our templates


 API demos

 API development

 API testing

 API security

 Incident response

 Cloud infrastructure management

Step 1 of 2

Cancel

Next

Postman: Workspace pixup

Create your workspace

Name

diplo-java-xx-jakarta

Summary

Who can access your workspace?

- ☒ Personal
Only you can access
- ☐ Private
Only invited team members can access
- ☐ Team
All team members can access
- ☐ Partner
Only invited partners and team members can access
- ☐ Public
Everyone can view

Step 2 of 2

Back

Create

Postman: Workspace pixup

 Overview



diplo-java-xx-jakarta

Workspace description

Add information that you want quick access to. It can include links to important resources or notes of what you want to remember.

Pinned collections



Make it easy to discover notable collections in this workspace by pinning them here.

[Explore workspace templates](#)

Configuración JAX-RS

```
import jakarta.ws.rs.ApplicationPath;  
import jakarta.ws.rs.core.Application;  
  
@ApplicationPath("api")  
public class ApplicationConfig extends Application {  
  
}
```

Capa API: TipoDomicilioApi

```
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
@Path("tiposDomicilio")
public interface TipoDomicilioApi {

    @GET
    Collection<TipoDomicilio> getAll();

}
```

Capa API: TipoDomicilioResource

```
public class TipoDomicilioResource implements TipoDomicilioApi {  
  
    @Inject  
    private TipoDomicilioRepository tipoDomicilioRepository;  
  
    @Override  
    public Collection<TipoDomicilio> getAll() {  
        return tipoDomicilioRepository.findAll();  
    }  
}
```


Capa API: ColoniaApi

```
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
@Path("colonias")
public interface ColoniaApi {

    @GET
    @Path("{id}")
    Colonia getColoniaById(@PathParam("id") Integer id);

    @GET
    Collection<Colonia> getColoniasByCp(
        @NotBlank @QueryParam("cp") String cp);
}
```

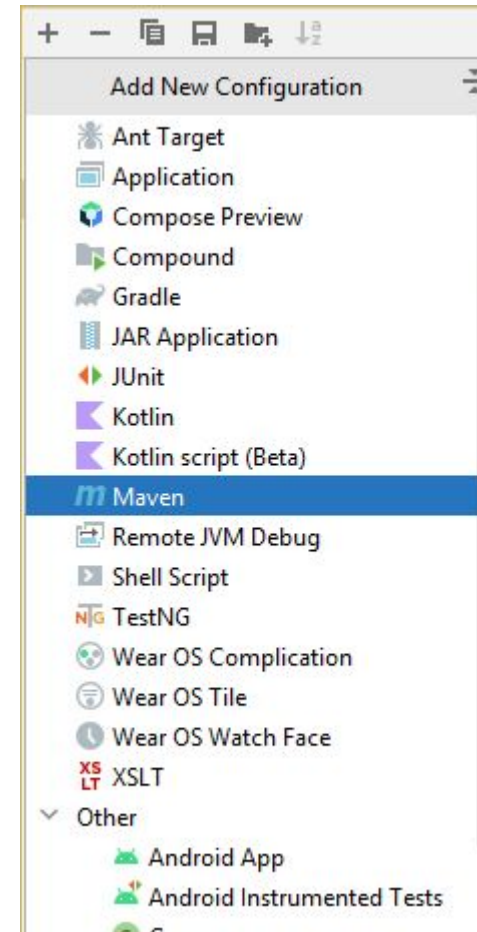
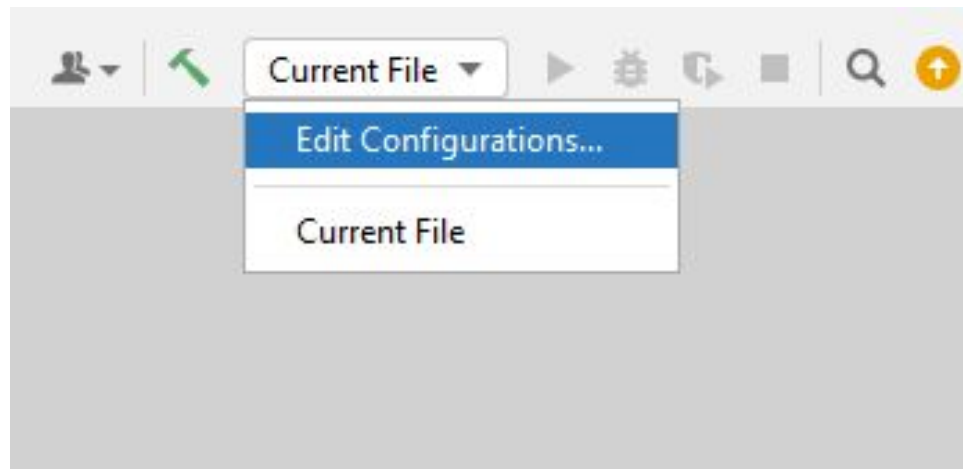
Capa API: ColoniaResource

```
public class ColoniaResource implements ColoniaApi {  
  
    @Inject  
    private ColoniaRepository coloniaRepository;  
  
    @Override  
    public Colonia getColoniaById(Integer id) {  
        Optional<Colonia> colonia = coloniaRepository.findById(id);  
        return colonia.orElse(null);  
    }  
  
    @Override  
    public Collection<Colonia> getColoniasByCp(String cp) {  
        return coloniaRepository.findByCp(cp);  
    }  
  
}
```

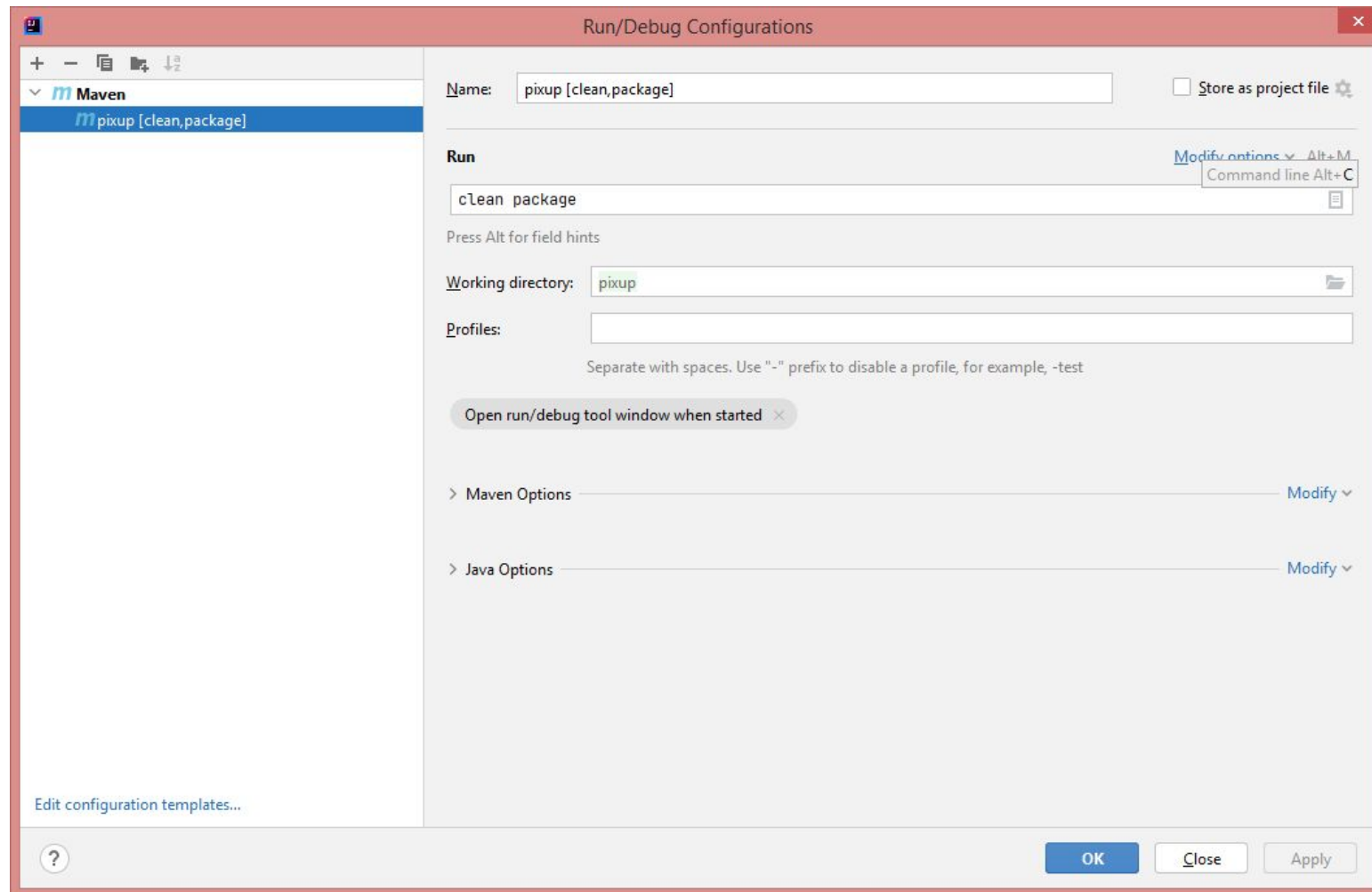


WildFly Application Server Deployment

Maven - Generar war



Maven - Generar war



Maven - Generar war

Recent Files [Ctrl+E](#)

Navigation Bar [Alt+Inicio](#)

Drop files here to open them

Run: [pixup \[clean,package\]](#) ×

5 sec, 798 ms

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 4.516 s  
[INFO] -----  
  
Process finished with exit code 0
```


Pixup Deployment WildFly





Deployments



A deployment represents anything that can be deployed (e.g. an application such as EJB-JAR, WAR, EAR, any kind of standard archive).

You can use **drag and drop** to add new content or replace existing deployments. Simply drag one or several files onto the deployment area. If a deployment with the same name already exists, it will be replaced, otherwise the deployment will be added. The deployments added by drag and drop will be enabled by default.

Pixup Deployment WildFly

Deployment (1)  




Filter by: name or deployment status

 pixup.war **View** 

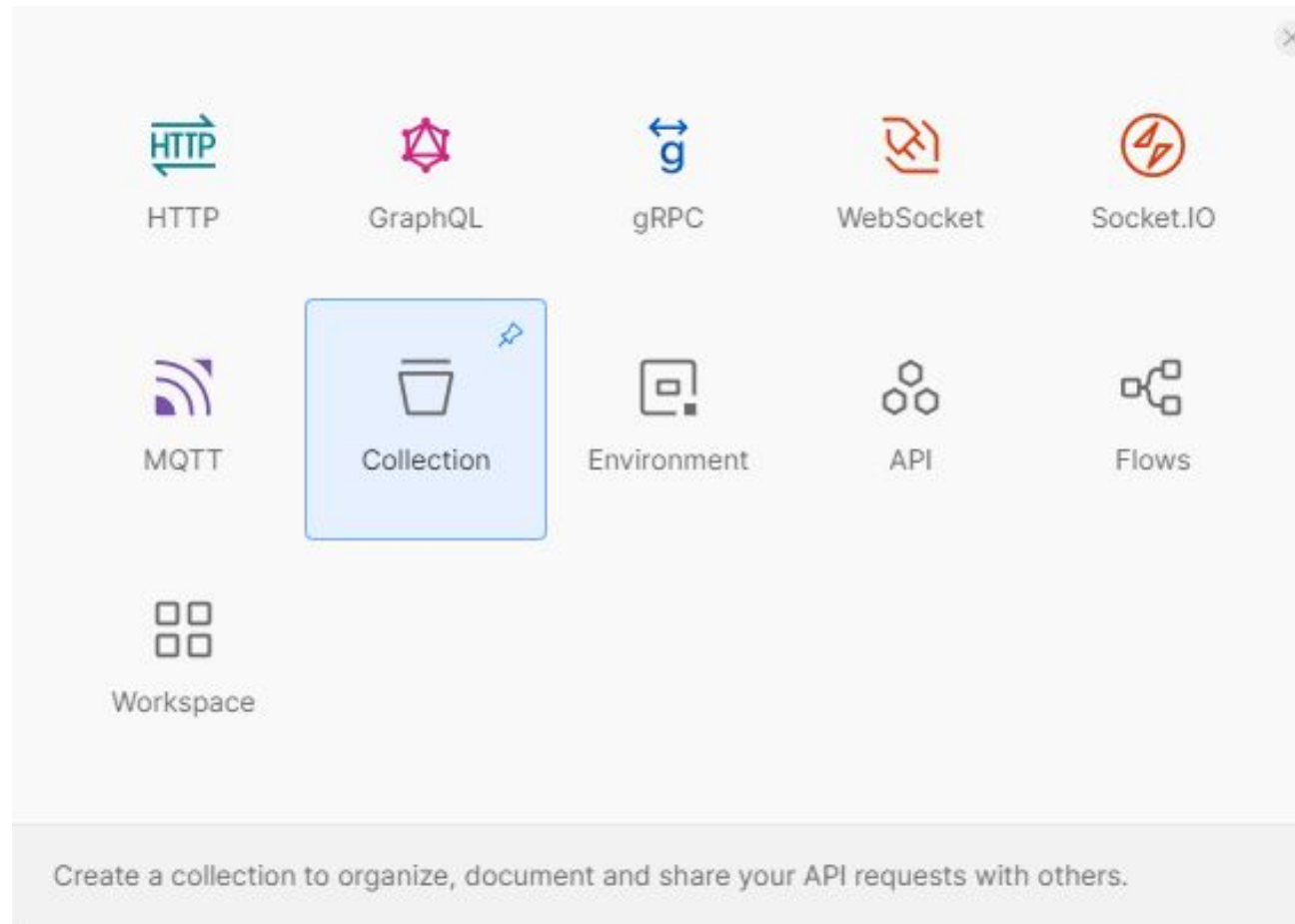
pixup.war

 The deployment **pixup.war** is enabled and active. [Disable](#)

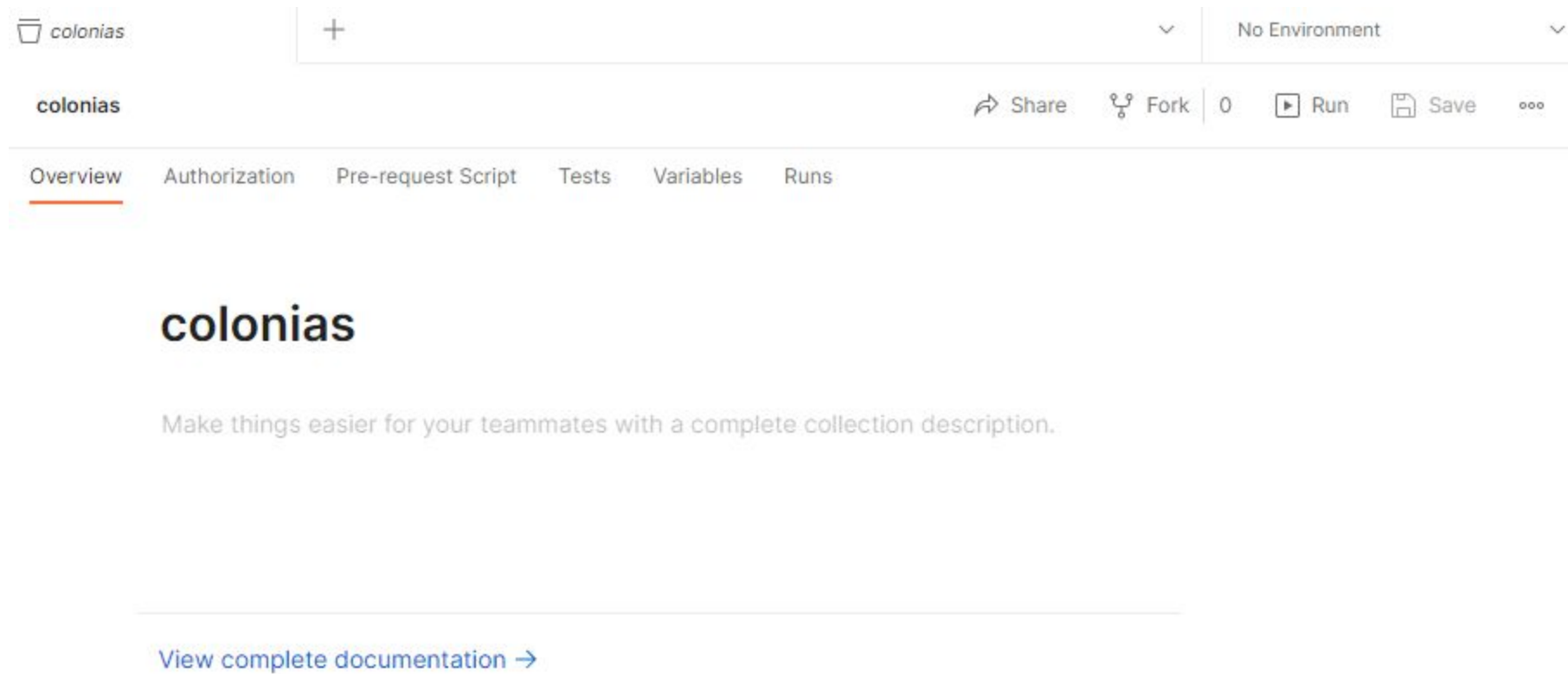
Main Attributes

Name:	pixup.war
Runtime Name:	pixup.war
Context Root:	/pixup
Hash:	f949bb4815fd0ed324f637e91adbeb7baa727b51
Enabled, Managed, Exploded:	  
Status:	OK
Last enabled at:	
Last disabled at:	n/a

Postman: Collection colonias



Postman: Collection colonias



Postman: ColoniaResource GET getColoniaById

The screenshot displays the Postman interface for a GET request. The request is named "getColoniaById" and is directed to the URL "http://localhost:8080/pixup/api/colonias/1". The response status is "200 OK" with a time of 748 ms and a size of 279 B. The response body is shown in JSON format, containing a list of colonies with their respective IDs, names, and state information.

Request Details:

- Method: GET
- URL: http://localhost:8080/pixup/api/colonias/1

Response Details:

- Status: 200 OK
- Time: 748 ms
- Size: 279 B

Response Body (JSON):

```
1 {
2   "cp": "06400",
3   "id": 1,
4   "municipio": {
5     "estado": {
6       "id": 1,
7       "nombre": "CIUDAD DE MÉXICO"
8     },
9     "id": 2,
10    "nombre": "Cuauhtémoc"
11  },
12  "nombre": "Santa María la Ribera"
13 }
```

Postman: ColoniaResource GET getColoniabyId - Not Found

The screenshot shows the Postman interface for a GET request. The request is named 'getColoniaById' and is located in the 'colonias' collection. The URL is 'http://localhost:8080/pixup/api/colonias/100'. The status of the request is '204 No Content', with a response time of 512 ms and a size of 64 B. The response body is empty, as indicated by the 'Status: 204 No Content' message and the 'Save as example' button.

GET getColoniaById

colonias / getColoniaById

GET http://localhost:8080/pixup/api/colonias/100

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (1) Test Results

Pretty Raw Preview Visualize Text

1

Status: 204 No Content Time: 512 ms Size: 64 B Save as example

Postman: ColoniaResource GET getColoniasbyCp

GET getColoniasByCp

colonias / getColoniasByCp

GET http://localhost:8080/pixup/api/colonias?cp=06400

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> cp	06400		
Key	Value	Description	

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 639 ms Size: 416 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "cp": "06400",
4     "id": 1,
5     "municipio": {
6       "estado": {
7         "id": 1,
8         "nombre": "CIUDAD DE MÉXICO"
9       },
10      "id": 2,
11      "nombre": "Cuahtémoc"
12    },
13    "nombre": "Santa María la Ribera"
14  },
15  {
16    "cp": "06400",
17    "id": 2,
18    "municipio": {
19      "estado": {
20        "id": 1,
21        "nombre": "CIUDAD DE MÉXICO"
22      },
23      "id": 2,
24      "nombre": "Cuahtémoc"
25    },
26    "nombre": "San Rafael"
27  }
28 }
```

Postman: ColoniaResource GET getColoniasbyCp - No Matches

GET getColoniasByCp

colonias / getColoniasByCp

GET http://localhost:8080/pixup/api/colonias?cp=15000

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	cp	15000			
	Key	Value	Description		

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

1 []

Status: 200 OK Time: 525 ms Size: 133 B Save as example

Postman: ColoniaResource GET getColoniasbyCp - Validator

The screenshot shows the Postman interface for a GET request to `http://localhost:8080/pixup/api/colonias?cp=`. The request is named `getColoniasByCp` and is part of a collection named `colonias`. The `Params` tab is active, showing a single query parameter `cp` with a checked checkbox. The `Body` tab is also active, showing a JSON body with parameter validation rules. The response status is `400 Bad Request` with a time of `531 ms` and a size of `236 B`.

Request Details:

- Method: GET
- URL: `http://localhost:8080/pixup/api/colonias?cp=`
- Environment: No Environment

Params:

Key	Value	Description
<input checked="" type="checkbox"/> cp		

Body:

```
[{"key": "[PARAMETER]", "value": "[getColoniasByCp.arg0]", "description": "[must not be blank]"}]
```

Response:

Status: 400 Bad Request | Time: 531 ms | Size: 236 B

Capa API: ColoniaResource Refactor Not Found Case

```
@Override
public Response getColoniaById(Integer id) {
    Optional<Colonia> colonia = coloniaRepository.findById(id);
    if (colonia.isPresent()) {
        return Response
            .status(Response.Status.OK)
            .entity(colonia.get())
            .build();
    }
    return Response
        .status(Response.Status.NOT_FOUND)
        .entity(null)
        .build();
}
```

Postman: ColoniaResource GET getColoniabyld - Not Found

The screenshot shows the Postman interface for a GET request. The request is named 'getColoniaByld' and is sent to 'http://localhost:8080/pixup/api/colonias/100'. The response status is '404 Not Found', with a time of 553 ms and a size of 106 B. The response body is empty, as indicated by the '1' in the body tab.

GET getColoniaByld

colonias / getColoniaByld

GET http://localhost:8080/pixup/api/colonias/100

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize Text

1

Status: 404 Not Found Time: 553 ms Size: 106 B Save as example

Postman: TipoDomicilioResource GET getAll

The screenshot shows the Postman interface for a GET request. The request is named 'getAll' and is sent to the URL 'http://localhost:8080/pixup/api/tiposDomicilio'. The response status is '200 OK' with a time of 557 ms and a size of 203 B. The response body is displayed in JSON format, showing a list of two objects representing different types of domiciles.

Request Details:

- Method: GET
- URL: http://localhost:8080/pixup/api/tiposDomicilio

Response Details:

Status: 200 OK, Time: 557 ms, Size: 203 B

Response Body (JSON):

```
[
  {
    "descripcion": "Entrega",
    "id": 1
  },
  {
    "descripcion": "Facturacion",
    "id": 2
  }
]
```




Stateless Session Bean ColoniaService

Stateless Session Bean

- Se utiliza la anotación **@Stateless**
- Deben contener un constructor sin argumentos para poderse instanciar.
- Un stateless session bean no retiene ningún estado (stateless - sin estado), lo que significa que las variables del bean no retienen ningún valor entre invocaciones.
- Un servidor de aplicaciones mantiene un pool de session beans para ser usado por sus clientes, cuando un cliente invoca un bean se toma desde el pool para su uso. Una vez concluida la invocación del cliente, el bean es regresado al pool, por lo que los beans son thread-safe y funcionan muy bien en entornos concurrentes.

Stateless Session Bean

- Pueden implementar una interface de negocio aunque no es requerido. Las interfaces pueden ser local o remota.
- **@Local**: Definir una interface local, usada por clientes que existen **dentro** del mismo contenedor de beans.
- **@Remote**: Definir una interface remota, usada por clientes que existen **fuera** del mismo contenedor de beans.

Stateless Session Bean - Ciclo de Vida

1. El contenedor crea un stateless session bean usando el constructor sin argumentos.
2. Las dependencias del bean son inyectadas.
3. Se crea un pool administrado de beans y múltiples instancias del session bean se ponen en el pool.
4. Cuando un cliente realiza un request se toma un bean inactivo del pool. Si todos los beans del pool se encuentran actualmente en uso, se crean más instancias del bean hasta alcanzar la cantidad máxima especificada.
5. El método de negocio invocado por el cliente es ejecutado.
6. El bean regresa al pool después de que la ejecución ha concluido.
7. El bean es destruido por el pool según sea necesario.

Capa de Persistencia: ColoniaRepository

```
public interface ColoniaRepository {  
  
    Collection<Colonia> findByCp(String cp);  
    Optional<Colonia> findById(Integer id);  
    Colonia saveOrUpdate(Colonia colonia);  
    void delete(Colonia colonia);  
    Optional<Colonia> findByCpAndNombre(String cp, String nombre);  
  
}
```

Capa de Negocio: ColoniaService

```
@Local
public interface ColoniaService {

    Colonia obtenerColoniaPorId(Integer id);
    Colonia crearColonia(Colonia colonia);
    Colonia actualizarColonia(Colonia colonia);
    void eliminarColoniaPorId(Integer id);

}
```

Capa de Negocio: ColoniaServiceImpl

```
@Stateless
public class ColoniaServiceImpl implements ColoniaService {

    @Inject
    private ColoniaRepository coloniaRepository;
    @Inject
    private MunicipioRepository municipioRepository;

    @Override
    public Colonia obtenerColoniaPorId(Integer id) {
        Optional<Colonia> colonia = coloniaRepository.findById(id);
        if (colonia.isPresent()) {
            return colonia.get();
        }
        throw new ColoniaNotFoundException(id);
    }
}
```


Capa API: ColoniaResource

```
@Override
public Response getColoniaById(Integer id) {
    try {
        Colonia colonia = coloniaService.obtenerColoniaPorId(id);
        return Response
            .status(Response.Status.OK)
            .entity(colonia)
            .build();
    } catch (Exception e) {
        return Response
            .status(Response.Status.NOT_FOUND)
            .entity(e.getCause().getMessage())
            .build();
    }
}
```

Postman: ColoniaResource GET getColoniabyId - Not Found

GET getColoniaById

colonias / getColoniaById

GET http://localhost:8080/pixup/api/colonias/100

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

1 No se encontró la colonia con id: 100

Status: 404 Not Found Time: 544 ms Size: 177 B Save as example

Contacto

Uriel Hernández
Solution Architect

urielhdezorozco@yahoo.com.mx

Redes sociales:

<https://www.linkedin.com/in/juho-mex>