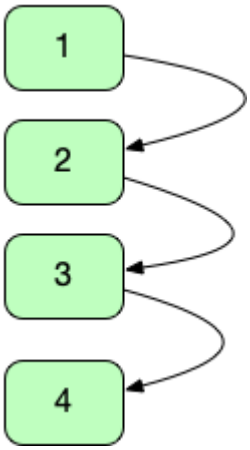
¿Que es un UUID? . En muchas ocasiones en Java necesitamos identificadores únicos . La la mayor parte de las veces suelen estar ligados a bases de datos y a claves primarias de la tabla. Es decir una tabla en su clave primaria necesita un identificador único y lo definimos de forma incremental. Hasta ahí todo correcto y desde una tecnología como JPA podríamos hacer lo siguiente:

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Usuario {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String nombre;
    private String apellido;

    ...etc
}
```

Esto hará que tengamos una clave única para cada registro de la tabla que se va incrementando paulatinamente según insertamos nuevos registros.



UUID al rescate

Sin embargo hay situaciones en las cuales puede que este valor incremental no sea suficiente ya que queramos un valor más universal. Por ejemplo puede ser que no nos interese que una persona sepa que la siguiente factura es la 20 si el tiene la 19 . En este caso el usar UUIDs nos puede ayudar los UUIDs son identificadores universales que se componen de una secuencia alfanumerica.

```
0181b9de-2a43-40cb-a2fc-ca58b62f4658
```

Pueden ser generados desde Java de una forma sencilla:

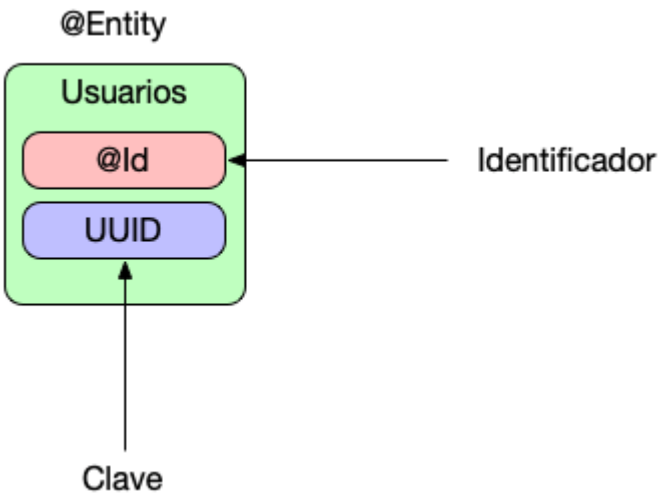
```
package com.arquitecturajava;

import java.util.UUID;

public class Principal {

	public static void main(String[] args) {

		UUID uuid= UUID.randomUUID();
		System.out.println(uuid);
	}
```

```
}
```

El resultado se imprime

```
0181b9de-2a43-40cb-a2fc-ca58b62f4658
```

Este identificador es único y podemos estar seguros de ello para la aplicación que nosotros manejemos . Los UUIDs se pueden usar también en JPA para identificadr de forma única elementos en una tabla.




```
@Entity
public class Usuario {

    @Id
    private UUID id;

    private String nombre;
    private String apellido;

    // Getters y setters, constructores y otros métodos aquí...

    // Resto de la clase...
```

```
}
```

Los UUId nos pueden ser útiles para situaciones de este estilo en las que no queremos publicar con claridad que existe una correspondencia numérica entre los distintos elementos a nivel de clave primaria

- Java List to Map y el uso de Collectors
- ¿Cuales son las certificaciones Java?
- Spring Security Annotation y su configuración
- StringBuffer, String y rendimiento
- JBoss Modules