

# Código Fuente

```
JpaLineaTelefonicaDao.java
78  * @author Andres Cantos R.
79  *
80  */
81  @Repository(value="lineatelefonica dao")
82  public class JpaLineaTelefonicaDao implements LineaTelefonicaDao {
83
84      private EntityManager em = null;
85
86      @Value("${spring.profiles.active}")
87      private String ambienteejecucion;
88
89
90      @PersistenceContext
91      public void setEntityManager(EntityManager em) {
92          this.em = em;
93      }
94
95      /**
96       * Este metodo consulta la disponibilidad de una linea telefonica segun el patron enviado como parametro
97       * para que la linea telefonica este libre, debe estar en estado lib
98       * devuelve
99       * a la fecha del 30 julio 2018, solo esta devolviendo hasta 20 registros por cada consulta
100      * @param prefijociudad Utilizar solo caracter ejemplo 2 (PICHINCHA), 4 (GUAYAS), 5 (MANABI)
101      * @param patronbusqueda texto utilizando el * como comodin, Ejemplo: todas las lineas que comiencen con 390* o lineas te terminen en *87
102      * @param empresa solicitante Ejemplo: TELCONET / NETVOICE / NETLIFE
103      */
104      @SuppressWarnings("unchecked")
105      @Override
106      @Transactional(readOnly=true)
107      public List<Pojo_NumeroTelefonico> consultadisponibilidadnumero(String prefijociudad, String patronbusqueda, String empresasolicitante, String canton
108      throws ServiciosNetvoiceExcepcion{
109          List<Pojo_NumeroTelefonico> lista = new LinkedList<>();
110
111          TblFaZona objzona = null; //Aquí voy a guardar el canton que recibo como parametro, si es que lo encuentro.
112
113          //cambio 13 de diciembre, consulto si tengo el canton recibido
114          try {
115              if(canton != null && !canton.trim().equals("")) {
116                  Query queryzone = em.createNamedQuery("TblFaZona.findxnombre", TblFaZona.class);
117                  queryzone.setParameter("nombre", "%"+canton+"%");
118                  List<TblFaZona> listzone = queryzone.getResultList();
119                  if(listzone!=null && listzone.size()>0) {
120                      objzona = listzone.get(0);
121                  }
122              }
123          } catch (NoResultException e) {
124              //No se encontro una zona para el texto recibido, no se hace nada.
125          }
126
127          try {
128              StoredProcedureQuery storedProcedure = em.createStoredProcedureQuery("[LINKOCORE].[dbo].[SP_CI_CONSULTARNUMEROSXPATRON]");
129              storedProcedure.registerStoredProcedureParameter(1,String.class,ParameterMode.IN);
130              storedProcedure.registerStoredProcedureParameter(2,String.class,ParameterMode.IN);
131              storedProcedure.registerStoredProcedureParameter(3,String.class,ParameterMode.IN);
132          }
```

```
RecaudacionesR...  RecaudacionesS...  GeneralesReposi...  TblCbRelCobroT...  DetCobroDctosP...  CamaleonWsExce...
1  package ec.netvoice.camaleon.excepciones;
2
3  /**
4   *
5   * @author Andres Cantos
6   * Creado: 12 Agosto 2020
7   * Clase encargada de manejar los errores que se presentan a nivel de Servicios para luego
8   * ser devueltos como un json al usuario.
9   */
10 public class CamaleonWsExcepcion extends IllegalArgumentException {
11
12     private static final long serialVersionUID = 4658281566898620259L;
13
14     private String codigoerror = "";
15     private String tipoerror = "";
16     private String mensaje = "";
17     private String trace = "";
18
19     public CamaleonWsExcepcion();
20     public CamaleonWsExcepcion(String msg){this.mensaje = msg; this.tipoerror="ERROR";};
21     public CamaleonWsExcepcion(String msg, String traza){this.mensaje = msg; this.trace = traza; this.tipoerror = "ERROR";};
22
23     /**
24      * Este método tiene agregado el tipo de error para los servicios web que necesitan mostrar un tipo específico en los
25      * json de error devueltos.
26      * @param msg mensaje de error
27      * @param traza traza de error provista por las excepciones
28      * @param tipoerror String que indica el tipo de error
29      */
30     public CamaleonWsExcepcion(String msg, String traza, String tipoerror){this.mensaje = msg; this.trace = traza; this.tipoerror = tipoerror;};
31
32     @Override
33     public String getMessage() {
34         return this.mensaje;
35     }
36
37     public String getMensaje(){
38         return this.mensaje;
39     }
40     public String getTrace() {
41         return trace;
42     }
43     public void setTrace(String traza) {
44         this.trace = traza;
45     }
46 }
```