**Logic Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | ERTA | Date | 13/11/2018 |
| Program | Clase SC\_Lectura | Program # | 2 |
| Instructor |  | Language | Java |

|  |  |
| --- | --- |
| Design |  |
| References | Functional Template.docx |
|  | Operational Specification Template.docx |
|  |  |
| Parameters |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
|  |
| Metodo public void RecibirMensaje (String NomCola, int id) Comienzo de metodo              Fábrica de ConnectionFactory = nueva ConnectionFactory ();              factory.setUri ("Url servidor de colas");              Conexión conexión = factory.newConnection ();              Canal del canal = crearCanal de conexion ();              Declarar la cola en del canal (NomCola, false, false, false, null);             imprimir ("[\*] En espera de mensajes. Para salir, presione CTRL + C");              Consumidor = nuevo DefaultConsumer (canal) {                 Metodo  publico void handleDelivery (String consumerTag, Envolvente del sobre, propiedades AMQP.BasicProperties, byte [] cuerpo)                          lanza IOException {                      Mensaje de cadena = nueva cadena (cuerpo, "UTF-8");                      Imprime ("[x] Received '" + message + "'");                      Cadena [] Fecha = message.split ("");                      Cadena fecha = "";                      para (int i = 0; i <Fecha.length; i ++) {                          fecha = Fecha [i];                      }                      Intancia de BD\_Usuario mBD = nuevo BD\_Usuario ();                          Llamar al metodo mBD.Conectar ();                          si (! Mandar llamar a BD método InsertarMsj (id, mensaje, fecha)) {                            Imprime ("Error");                          }                        }                  }              }   Cierre del método |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |