

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Ingeniería en Ciencias y Sistemas
Introducción a la Programación de Computadoras 1
Ing. William Estuardo Escobar Argueta
Freddy Alejandro Monterroso



Proyecto 1

USAC - Delivery

Enrique Fernando Gaitán Ibarra
202110531

Nueva Guatemala de la Asunción, 10 de marzo de 2023

Contenido

Usuarios.....	3
Información.....	3
Código.....	3
Ubicaciones	5
Información.....	5
Código.....	5
Datos de Facturación	7
Información.....	7
Código.....	7
Compras	9
Información.....	9
Código.....	9
Tools.....	10
Información.....	10
Código.....	10

Package Controlador

En este paquete se encuentran contenidas todas las clases que se utilizan en el proyecto.

Usuarios

Información

Clases: Autenticador, User

Uso: Estas son las clases que se utilizan para el registro de usuarios.

Estructura: La clase Autenticador contiene los atributos de correo electrónico, contraseña y rol que se heredan a la clase User, que es donde están los atributos de información personal. Cada clase contiene los métodos set y get de sus atributos.

Código

```
1  public class Autenticador {
2
3      private String correo;
4      private String password;
5      private String rol;
6
7      public Autenticador(String correo, String password, String rol){
8          this.correo = correo;
9          this.password = password;
10         this.rol = rol;
11     }
12
13     public String getCorreo() {
14         return correo;
15     }
16
17     public String getPassword() {
18         return password;
19     }
20
21     public String getRol() {
22         return rol;
23     }
24
25     public void setCorreo(String correo) {
26         this.correo = correo;
27     }
28
29     public void setPassword(String password) {
30         this.password = password;
31     }
32
33     public void setRol(String rol) {
34         this.rol = rol;
35     }
36 }
37
38
```



```
1  public class Autenticador {
2
3      private String correo;
4      private String password;
5      private String rol;
6
7      public Autenticador(String correo, String password, String rol){
8          this.correo = correo;
9          this.password = password;
10         this.rol = rol;
11     }
12
13     public String getCorreo() {
14         return correo;
15     }
16
17     public String getPassword() {
18         return password;
19     }
20
21     public String getRol() {
22         return rol;
23     }
24
25     public void setCorreo(String correo) {
26         this.correo = correo;
27     }
28
29     public void setPassword(String password) {
30         this.password = password;
31     }
32
33     public void setRol(String rol) {
34         this.rol = rol;
35     }
36
37 }
38
```

Ubicaciones

Información

Clases: Region, Departamento, Municipio, Kiosco

Uso: Estas son las clases que se utilizan para el registro de las ubicaciones en donde se realizarán las entregas.

Estructura: La clase Region es la clase padre, mientras que la clase Departamento y Kiosco contiene a la clase Region como subhijas y lo mismo sucede con Municipio y Departamento. No sucede por medio de herencia. Cada clase contiene los métodos set y get de sus atributos.

Código

```
1  public class Region {
2      private String codigo;
3      private String nombre;
4      private double priceE;
5      private double priceS;
6
7      public Region(String codigo, String nombre, double priceE, double priceS) {
8          this.codigo = codigo;
9          this.nombre = nombre;
10         this.priceE = priceE;
11         this.priceS = priceS;
12     }
13
14     public String getCodigo() {
15         return codigo;
16     }
17
18     public void setCodigo(String codigo) {
19         this.codigo = codigo;
20     }
21
22     public String getNombre() {
23         return nombre;
24     }
25
26     public void setNombre(String nombre) {
27         this.nombre = nombre;
28     }
29
30     public double getPriceE() {
31         return priceE;
32     }
33
34     public void setPriceE(double priceE) {
35         this.priceE = priceE;
36     }
37
38     public double getPriceS() {
39         return priceS;
40     }
41
42     public void setPriceS(double priceS) {
43         this.priceS = priceS;
44     }
45 }
```

```
1 public class Departamento {
2
3     private String cod;
4     private String name;
5     private Region reg;
6
7     public Departamento(String cod, String name, Region reg) {
8         this.cod = cod;
9         this.name = name;
10        this.reg = reg;
11    }
12
13    public String getCod() {
14        return cod;
15    }
16
17    public void setCod(String cod) {
18        this.cod = cod;
19    }
20
21    public String getName() {
22        return name;
23    }
24
25    public void setName(String name) {
26        this.name = name;
27    }
28
29    public Region getReg() {
30        return reg;
31    }
32
33    public void setReg(Region reg) {
34        this.reg = reg;
35    }
36
37 }
```

```
1 public class Kiosco {
2
3     private String codigo;
4     private String Nombre;
5     private Region reg;
6
7     public Kiosco(String codigo, String Nombre, Region reg) {
8         this.codigo = codigo;
9         this.Nombre = Nombre;
10        this.reg = reg;
11    }
12
13    public String getCodigo() {
14        return codigo;
15    }
16
17    public void setCodigo(String codigo) {
18        this.codigo = codigo;
19    }
20
21    public String getNombre() {
22        return Nombre;
23    }
24
25    public void setNombre(String Nombre) {
26        this.Nombre = Nombre;
27    }
28
29    public Region getReg() {
30        return reg;
31    }
32
33    public void setReg(Region reg) {
34        this.reg = reg;
35    }
36
37
38 }
```

Datos de Facturación

Información

Clases: Factura, Tarjeta

Uso: Estas son las clases que se utilizan para que cada usuario registre la información de facturación que se muestra en cada compra.

Estructura: La clase Factura tiene como atributos los datos principales que debe incluir una factura (nombre, dirección y nit) y la clase Tarjeta tiene los atributos que una tarjeta tiene como información principal (nombre, numeración y fecha de vencimiento). Cada clase contiene los métodos set y get de sus atributos.

Código

```
1  public class Factura {
2      private String nombre;
3      private String direccion;
4      private String nit;
5      private String dpi;
6
7      public Factura(String nombre, String direccion, String nit, String dpi) {
8          this.nombre = nombre;
9          this.direccion = direccion;
10         this.nit = nit;
11         this.dpi = dpi;
12     }
13
14     public String getNombre() {
15         return nombre;
16     }
17
18     public void setNombre(String nombre) {
19         this.nombre = nombre;
20     }
21
22     public String getDireccion() {
23         return direccion;
24     }
25
26     public void setDireccion(String direccion) {
27         this.direccion = direccion;
28     }
29
30     public String getNit() {
31         return nit;
32     }
33
34     public void setNit(String nit) {
35         this.nit = nit;
36     }
37
38     public String getDpi() {
39         return dpi;
40     }
41
42     public void setDpi(String dpi) {
43         this.dpi = dpi;
44     }
45 }
```

La clase Tarjeta contiene un algoritmo en el método getNumero, debido a que los primeros 12 dígitos de una tarjeta de deben ser ocultados, por cuestión de seguridad.

```
1  public class Tarjeta {
2      private String nombre;
3      private String numeroT;
4      private String fechaVen;
5      private String dpi;
6
7      public Tarjeta(String nombre, String numeroT, String fechaVen, String dpi) {
8          this.nombre = nombre;
9          this.numeroT = numeroT;
10         this.fechaVen = fechaVen;
11         this.dpi = dpi;
12     }
13
14     public String getNombre() {
15         return nombre;
16     }
17
18     public void setNombre(String nombre) {
19         this.nombre = nombre;
20     }
21
22     public String getNumeroT() {
23         String str = numeroT;
24         int pos = numeroT.length()-4;
25         String prt1 = str.substring(0, pos-1);
26         String prt2 = str.substring(pos);
27
28         int largo = prt1.length();
29
30
31         char ch = '*';
32         char[] chars = prt1.toCharArray();
33
34         for(int i=0; i<largo; i++){
35             chars[i] = ch;
36         }
37
38         prt1 = String.valueOf(chars);
39
40         numeroT = prt1 + prt2;
41
42         return numeroT;
43     }
44
45     public void setNumeroT(String numeroT) {
46         this.numeroT = numeroT;
47     }
48
49     public String getFechaVen() {
50         return fechaVen;
51     }
52
53     public void setFechaVen(String fechaVen) {
54         this.fechaVen = fechaVen;
55     }
56
57     public String getDpi() {
58         return dpi;
59     }
60
61     public void setDpi(String dpi) {
62         this.dpi = dpi;
63     }
64
65 }
66
67
```


Compras

Información

Clases: Cotizacion, compra

Uso: Estas son las clases que se utilizan para que el pago de envíos.

Estructura: La clase compra registra todos los datos que se obtienen cuando se hará un envío, así mismo, es la clase donde más se incluye a otras como Departamento, Municipio y Región, las cuales se presentan en los JComboBox, al igual que Facturas y Tarjetas. Cada clase contiene los métodos set y get de sus atributos

Código

```
1 public class Compra extends Cotizacion {
2
3     private String cod;
4     private double total;
5     private String typePay;
6     private String facturacion;
7     private String dpi;
8     private String typeSend;
9     private String destinatario;
10
11     public Compra(String cod, double total, String typePay, String facturacion, String dpi, String typeSend, String destinatario, Departamento departamento0, Municipio municipio0, String direccion0, Departamento departamento0, Municipio municipio0, String direccion0, int cantidadPaquetes, String sizepackage) {
12         super(departamento0, municipio0, direccion0, departamento0, municipio0, direccion0, cantidadPaquetes, sizepackage);
13         this.cod = cod;
14         this.total = total;
15         this.typePay = typePay;
16         this.facturacion = facturacion;
17         this.dpi = dpi;
18         this.typeSend = typeSend;
19         this.destinatario = destinatario;
20     }
21
22     public String getCod() {
23         return cod;
24     }
25
26     public void setCod(String cod) {
27         this.cod = cod;
28     }
29
30     public double getTotal() {
31         return total;
32     }
33
34     public void setTotal(double total) {
35         this.total = total;
36     }
37
38     public String getTypePay() {
39         return typePay;
40     }
41
42     public void setTypePay(String typePay) {
43         this.typePay = typePay;
44     }
45
46     public String getFacturacion() {
47         return facturacion;
48     }
49
50     public void setFacturacion(String facturacion) {
51         this.facturacion = facturacion;
52     }
53
54     public String getDpi() {
55         return dpi;
56     }
57
58     public void setDpi(String dpi) {
59         this.dpi = dpi;
60     }
61
62     public String getTypeSend() {
63         return typeSend;
64     }
65
66     public void setTypeSend(String typeSend) {
67         this.typeSend = typeSend;
68     }
69
70     public String getDestinatario() {
71         return destinatario;
72     }
73
74     public void setDestinatario(String destinatario) {
75         this.destinatario = destinatario;
76     }
77 }
78
```

Tools

Información

Clases: RegionSends, UserSends, Utils

Uso: Estas clases son usadas como listas para almacenar información que se muestran en los reportes por parte del usuario administrador. La clase Utils contiene como atributos todos los linklist, así como, los método de visualización y creación de html para factura y guía.

Código

```
1  public class UserSends {
2      private String name;
3      private String apellido;
4      private String dpi;
5      private int cantidad;
6
7      public UserSends(String name, String apellido, String dpi, int cantidad) {
8          this.name = name;
9          this.apellido = apellido;
10         this.dpi = dpi;
11         this.cantidad = cantidad;
12     }
13
14     public String getName() {
15         return name;
16     }
17
18     public void setName(String name) {
19         this.name = name;
20     }
21
22     public String getApellido() {
23         return apellido;
24     }
25
26     public void setApellido(String apellido) {
27         this.apellido = apellido;
28     }
29
30     public String getDpi() {
31         return dpi;
32     }
33
34     public void setDpi(String dpi) {
35         this.dpi = dpi;
36     }
37
38     public int getCantidad() {
39         return cantidad;
40     }
41
42     public void setCantidad(int cantidad) {
43         this.cantidad = cantidad;
44     }
45 }
46
47
```



```
1  public class UserSends {
2      private String name;
3      private String apellido;
4      private String dpi;
5      private int cantidad;
6
7      public UserSends(String name, String apellido, String dpi, int cantidad) {
8          this.name = name;
9          this.apellido = apellido;
10         this.dpi = dpi;
11         this.cantidad = cantidad;
12     }
13
14     public String getName() {
15         return name;
16     }
17
18     public void setName(String name) {
19         this.name = name;
20     }
21
22     public String getApellido() {
23         return apellido;
24     }
25
26     public void setApellido(String apellido) {
27         this.apellido = apellido;
28     }
29
30     public String getDpi() {
31         return dpi;
32     }
33
34     public void setDpi(String dpi) {
35         this.dpi = dpi;
36     }
37
38     public int getCantidad() {
39         return cantidad;
40     }
41
42     public void setCantidad(int cantidad) {
43         this.cantidad = cantidad;
44     }
45 }
46
47
```

```

1  public class Utils {
2
3      public static LinkedList<Compra> listSolds = new LinkedList<>();
4      public static LinkedList<Departamento> listDepartamento = new LinkedList<>();
5      public static LinkedList<Factura> listFactura = new LinkedList<>();
6      public static LinkedList<Kiosco> listKiosco = new LinkedList<>();
7      public static LinkedList<Municipio> listMucipio = new LinkedList<>();
8      public static LinkedList<Region> listReg = new LinkedList<>();
9      public static LinkedList<RegionSends> listRegionSolds = new LinkedList<>();
10     public static LinkedList<Tarjeta> listCard = new LinkedList<>();
11     public static LinkedList<User> listUser = new LinkedList<>();
12     public static LinkedList<UserSends> listUserSolds = new LinkedList<>();

```

```

1  public static void regionSendsOrder(LinkedList<RegionSends> listRegionSolds, LinkedList<Compra> listSolds, LinkedList<Region> listReg) {
2
3      for (int i = 0; i < listReg.size(); i++) {
4          RegionSends rS = new RegionSends(listReg.get(i).getNombre(), 0);
5          listRegionSolds.add(rS);
6      }
7
8      for (int i = 0; i < listSolds.size(); i++) {
9          for (int j = 0; j < listRegionSolds.size(); j++) {
10             if (listSolds.get(i).getDepartamentoD().getReg().getNombre().equals(listRegionSolds.get(j).getReg())) {
11                 int cantidad = listRegionSolds.get(j).getCantidad() + 1;
12                 listRegionSolds.get(j).setCantidad(cantidad);
13             }
14         }
15     }
16 }
17
18 public static void userSendsOrder(LinkedList<UserSends> listUserSolds, LinkedList<Compra> listSolds, LinkedList<User> listUser) {
19
20     for (int i = 0; i < listUser.size(); i++) {
21         UserSends uS = new UserSends(listUser.get(i).getNombre(), listUser.get(i).getApellido(), listUser.get(i).getDpi(), 0);
22         listUserSolds.add(uS);
23     }
24
25     for (int i = 0; i < listSolds.size(); i++) {
26         for (int j = 0; j < listUserSolds.size(); j++) {
27             if (listSolds.get(i).getDpi().equals(listUserSolds.get(j).getDpi())) {
28                 int cantidad = listUserSolds.get(j).getCantidad() + 1;
29                 listUserSolds.get(j).setCantidad(cantidad);
30             }
31         }
32     }
33 }
34
35 }

```

```

1 public static String headCheck(String cod, LinkedList<Compra> listSolds, LinkedList<User> listUser, LinkedList<Factura> listFactura) {
2     for (int i = 0; i < listSolds.size(); i++) {
3         String munda = "";
4         String address = "";
5         for (int i = 0; i < listSolds.size(); i++) {
6             if (listSolds.get(i).getCod().equals(cod)) {
7                 index = i;
8             }
9         }
10        for (int i = 0; i < listUser.size(); i++) {
11            if (listUser.get(i).getPci().equals(listSolds.get(index).getPci())) {
12                munda = listUser.get(i).getNombre() + " " + listUser.get(i).getApellido();
13            }
14        }
15        for (int i = 0; i < listFactura.size(); i++) {
16            if (listFactura.get(i).getPci().equals(listSolds.get(index).getPci())) {
17                address = listFactura.get(i).getDireccion();
18            }
19        }
20    }
21    String encabezado = "<doctype html>\n"
22    + "<html lang='es'>\n"
23    + "<head>\n"
24    + "    <meta charset='UTF-8'>\n"
25    + "    <meta name='description' content='FACTURA'>\n"
26    + "    <meta name='robots' content='index, follow'>\n"
27    + "    <meta name='viewport' content='width=device-width, initial-scale=1.0'>\n"
28    + "    <title>FACTURA</title>\n"
29    + "    <style type='text/css' media='screen'>\n"
30    + "    {\n"
31    + "        box-sizing: border-box;\n"
32    + "        margin: 0;\n"
33    + "        padding: 0;\n"
34    + "        font-family: Arial, Helvetica, sans-serif;\n"
35    + "    }\n"
36    + "    html{\n"
37    + "        font-size: 62.5%;\n"
38    + "    }\n"
39    + "    body{\n"
40    + "        font-size: 1.6em;\n"
41    + "    }\n"
42    + "    header{\n"
43    + "        display: flex;\n"
44    + "        justify-content: center;\n"
45    + "    }\n"
46    + "    table{\n"
47    + "        margin: 10px;\n"
48    + "        padding: 5px;\n"
49    + "        border-spacing: 5px;\n"
50    + "        border-collapse: collapse;\n"
51    + "    }\n"
52    + "    .tableCheck {\n"
53    + "        width: 100%;\n"
54    + "    }\n"
55    + "    .titleTable {\n"
56    + "        text-align: center;\n"
57    + "        color: #000000;\n"
58    + "        font-size: 1.5em;\n"
59    + "    }\n"
60    + "    .infoTable {\n"
61    + "        text-align: left;\n"
62    + "    }\n"
63    + "    .ingTable {\n"
64    + "        text-align: center;\n"
65    + "    }\n"
66    + "    main {\n"
67    + "        display: flex;\n"
68    + "        justify-content: center;\n"
69    + "    }\n"
70    + "    .productTable{\n"
71    + "        border-collapse: collapse;\n"
72    + "        margin: 20px 0;\n"
73    + "        min-width: 800px;\n"
74    + "        width: 50%;\n"
75    + "    }\n"
76    + "    .productTable thead tr {\n"
77    + "        background-color: #000000;\n"
78    + "        color: #ffffff;\n"
79    + "        text-align: center;\n"
80    + "    }\n"
81    + "    .productTable th, .productTable td {\n"
82    + "        padding: 12px 15px;\n"
83    + "    }\n"
84    + "    .productTable tbody tr {\n"
85    + "        border-bottom: 1px solid #000000;\n"
86    + "    }\n"
87    + "    .productTable tbody tr:nth-of-type(even) {\n"
88    + "        background-color: #f3f3f3;\n"
89    + "    }\n"
90    + "    footer{\n"
91    + "        display: flex;\n"
92    + "        justify-content: center;\n"
93    + "    }\n"
94    + "    </style>\n"
95    + "</head>\n"
96    + "    <body>\n"
97    + "        <div class='tableCheck'>\n"
98    + "            <tr>\n"
99    + "                <td class='titleTable'>NUMERO FACTURA</td>\n"
100    + "                <td class='infoTable'> - index - </td>\n"
101    + "                <td class='ingTable'>\n"
102    + "                    <figure>\n"
103    + "                        <img alt='LOGO USAC DELIVERY'>\n"
104    + "                    </figure>\n"
105    + "                </td>\n"
106    + "            </tr>\n"
107    + "            <tr>\n"
108    + "                <td class='titleTable'>CODIGO</td>\n"
109    + "                <td class='infoTable'> - listSolds.get(index).getCod() - </td>\n"
110    + "            </tr>\n"
111    + "            <tr>\n"
112    + "                <td class='titleTable'>ORIGEN</td>\n"
113    + "                <td class='infoTable'> - listSolds.get(index).getDepartamento().getNombre() + ", " + listSolds.get(index).getMunicipio().getNombre() + ", " + listSolds.get(index).getDireccion() + </td>\n"
114    + "                <td class='titleTable'>DESTINO</td>\n"
115    + "                <td class='infoTable'> - listSolds.get(index).getDepartamento().getNombre() + ", " + listSolds.get(index).getMunicipio().getNombre() + ", " + listSolds.get(index).getDireccion() + </td>\n"
116    + "            </tr>\n"
117    + "            <tr>\n"
118    + "                <td class='titleTable'>NIT</td>\n"
119    + "                <td class='infoTable'> - listSolds.get(index).getFacturacion() + </td>\n"
120    + "            </tr>\n"
121    + "            <tr>\n"
122    + "                <td class='titleTable'>TIPO DE PAGO</td>\n"
123    + "                <td class='infoTable'> - listSolds.get(index).getTipoPay() + </td>\n"
124    + "            </tr>\n"
125    + "            <tr>\n"
126    + "                <td class='titleTable'>NUMERO</td>\n"
127    + "                <td class='infoTable'> - munda + </td>\n"
128    + "            </tr>\n"
129    + "            <tr>\n"
130    + "                <td class='titleTable'>DIRECCION</td>\n"
131    + "                <td class='infoTable'> - address + </td>\n"
132    + "            </tr>\n"
133    + "        </div>\n"
134    + "        <table class='productTable'>\n"
135    + "            <thead>\n"
136    + "                <tr>\n"
137    + "                    <th>ITEM</th>\n"
138    + "                    <th>DESCRIPCION</th>\n"
139    + "                    <th>CANTIDAD</th>\n"
140    + "                    <th>PRECIO</th>\n"
141    + "                    <th>TOTAL</th>\n"
142    + "                </tr>\n"
143    + "            </thead>\n"
144    + "            <tbody>\n"
145    + "                <tr>\n"
146    + "                    <td> - listSolds.get(index).getSlapackage() + </td>\n"
147    + "                    <td> - listSolds.get(index).getCantidadPaquetes() + </td>\n"
148    + "                    <td> - listSolds.get(index).getTotal() + </td>\n"
149    + "                </tr>\n"
150    + "            </tbody>\n"
151    + "        </table>\n"
152    + "    </body>\n"
153    + "    </html>\n"
154    + "    return encabezado + body + footer;
155 }

```

