

Universidad Técnica Particular de Loja

Ciencia de la Computación. Arquitectura de Aplicaciones Evaluación 1er Parcial

Por: Ramiro Israel Vivanco Gualán

Paralelo: B

Fecha: 12/11/2020

Desarrollar el registro de cuentas de usuario de la plataforma web, para la administración de parqueamientos.

El registro se hace por tres medios:

- a) Registro de cuenta, con confirmación de password.
- b) Registro de cuenta, con alta de cuenta por parte de administrador.
- c) Reistro de cuenta, por confirmación de correo electrónicos.

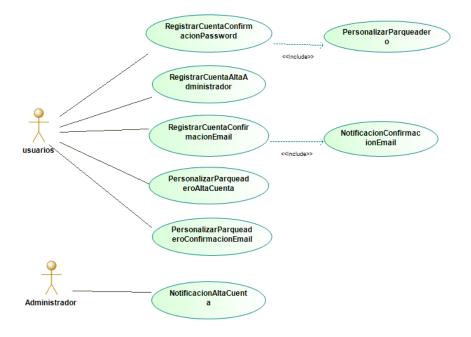
Registro de cuenta: nombreapellidos, fecha, login, password, email. Las cuentas no pueden registrarse si el login como el email, existen.

Luego de registrar la cuenta, la cuenta que debe registra los datos del parqueadero. Estas actividades se realizar en función de la siguiente eventualidad:

- a) Se hace en el momento de ingresar lo datos de la cuenta.
- b) Se hace después de que el administrador de la cuenta de alta a la cuenta
- c) Se hace después la confirmación por correo.
- d) Personalización de parqueadero: nombreempresa, direccionempresa, telefono, Esta personalización debe efectuarse obligatoriamente durante el registro

Desarrollar

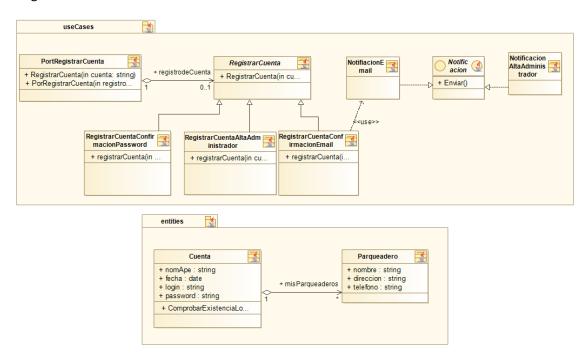
Diagrama de casos de uso





Universidad Técnica Particular de Loja Ciencia de la Computación. Arquitectura de Aplicaciones Evaluación 1er Parcial

Diagrama de clases.



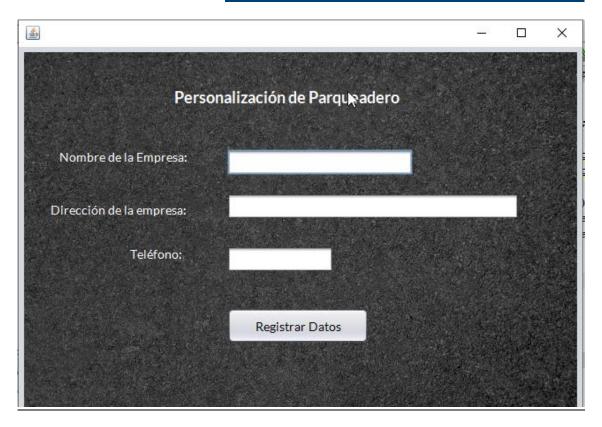
Código fuente que refleje el diagrama de clase.





Universidad Técnica Particular de Loja

Ciencia de la Computación. Arquitectura de Aplicaciones Evaluación 1er Parcial



GUI

```
y GOLIJAVA \land 📋 PELSONALIZACION PARKING JAVA \Lambda 🔯 KEYISU ALCUENTACONNI MACONI PASSWOYOLIJAVA 🐧 🔯 KEYISU ALCUENTACONNI MACONI MACONI PASSWOYOLIJAVA
Source Design History 🕼 🎜 🔻 🗒 🔻 💆 😓 😓 📮 📮 🎒 👙 😓 💆 🚇 🕮 📗 🏙 🚅
28 -
           public GUI() {
29
               initComponents();
30
31
32
   * This method is called from within the constructor to initialize the form.
33
34
            \ensuremath{^{*}} WARNING: Do NOT modify this code. The content of this method is always
            \ensuremath{^{\star}} regenerated by the Form Editor.
35
36
37
           @SuppressWarnings("unchecked")
38
   +
           Generated Code
133
134
           private void jNombreActionPerformed(java.awt.event.ActionEvent evt) {
                // TODO add your handling code here:
135
136
137
138
           private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {
139
               Cuenta objCuenta = new Cuenta();
140
               objCuenta.setNomApe(this.jNombre.getText().toString());
 9<u>.</u>
               objCuenta.setLogin(this.jLogin.getText().toString());
               objCuenta.setPassword(this.jPassword.getText().toString());
 <u>Q.</u>
               objCuenta.setRePassword(this.JRepass.getText().toString());
145
146
               PortRegistrarCuenta objPort = new PortRegistrarCuenta();
147
               objPort.PorRegistrarCuenta( new RegistrarCuentaConfirmacionEmail(objCuenta));
148
               objPort.PorRegistrarCuenta( new RegistrarCuentaConfirmacionPassword(objCuenta));
149
150
```

Universidad Técnica Particular de Loja

Ciencia de la Computación. Arquitectura de Aplicaciones Evaluación 1er Parcial

Verificar Pasword

```
1
     package useCases;
2
3
  import UI.GUI;
4
5
     import UI.PersonalizacionParking;
6
      import entities.Cuenta;
    import javax.swing.JOptionPane;
8
9
     public class RegistrarCuentaConfirmacionPassword extends RegistrarCuenta {
10
     PersonalizacionParking objPPark = new PersonalizacionParking();
     GUI objGUI = new GUI();
11
12
13
14 public RegistrarCuentaConfirmacionPassword(Cuenta objCuenta) {
15
16
             if(objCuenta.getPassword().equals(objCuenta.getRePassword()))
17
            JOptionPane.showMessageDialog(null, "Contraseñas coinciden");
18
            objPPark.setVisible(true);
19
20
            objGUI.setVisible(false);
21
             }else{
22
             JOptionPane.showMessageDialog(null, "Contraseñas no coinciden");
23
24
25
  뒨
26
         public void registrarCuenta(final String cuenta) {
27
28
```

Verificar Email

```
Source History | 🔀 🐶 🚚 - | 💽 👺 👺 🖶 🖫 | 🔗 😓 | 💇 💇 | 🧶 🔠 🕌 |
1
     package useCases;
2
3
  import entities.Cuenta;
     import javax.swing.JOptionPane;
5
8
9
     public class RegistrarCuentaConfirmacionEmail extends RegistrarCuenta {
10
11
  public RegistrarCuentaConfirmacionEmail(Cuenta ObjCuenta) {
12
13
              if(ObjCuenta.getLogin().equals("parking@email.com")){
                 JOptionPane. showMessageDialog(null, "El email ya existe");
14
15
             }else{
             JOptionPane.showMessageDialog(null, "El email válido");
16
17
18
19
          1
20
21
22
  무
23
         public void registrarCuenta(final Cuenta cuenta) {
24
25
26
27
```