

## Análisis de componentes de los datos de paciente

Equipo 1: Rafael Alejandro Maldonado Valle & Moises Vidal Hernández



31	jLabelTitulo2 (Historial Clínico)
42	JPanelDiseño4
43	JPanelDiseño5
51	jPanelDiseño13

### 31. JLabelTitulo2 (Historial Clínico)

Este componente que es una etiqueta nos mostrara el nombre de la ventana que es "Historial Clínico" en la parte superior de esta misma.

### 42. JPanelDiseño4

Solo es un panel de color lila que le da formato al encabezado de la parte izquierda de la ventana.

### 43. JPanelDiseño5

Solo es un panel de color lila que le da formato al encabezado de la parte derecha de la ventana.

### 51. JPanelDiseño13

Este es un panel que sirve para darle ese fondo color blanco a toda la ventana.

32	jLabelTitulo3 (Datos del paciente)
44	jPanelDiseño6
45	jPanelDiseño7
52	jLabelNomComPac
53	jTextFieldNomComPac
54	jLabelFolio
55	jTextFieldFolio
56	jLabelCarp2
57	jLTextFieldCarp2
58	jLabelFecNac
59	jTextFieldFecNac
60	jLabelTelefono
61	jTextFieldTelefono

### 31. JLabelTitulo3 (Datos del paciente)

Esta etiqueta nos mostrara un texto que dice “Datos Del Paciente”, esta nos indicara el apartado de los datos en el historial clínico correspondientes a un paciente en específico.

### 44. jPanelDiseño6

Panel que ayuda a resaltar y dar formato al JLabelTitulo3 en su parte superior.

### 45. jPanelDiseño7

Un panel que ayuda a resaltar y dar formato al JLabelTitulo3 en su parte inferior.

## 52. JLabelNomComPac

Esta etiqueta muestra un texto diciendo “Nombre Completo”, este mismo nos indicara en donde se insertará y se mostrara el nombre del paciente.

## 53. jTextFieldNomComPac

Es una caja de texto que nos permitirá guardar, capturar y mostrar el nombre del paciente.

Evaluaremos en base a 2 casos de uso

### ➤ Caso 1

Cuando no se encuentra el nombre en la base de datos, si se da clic en el botón consultar, aquí emergerá una ventana diciendo que no se encontró registro en base de datos, se nos mostrara una opción de agregar, entonces se toman los datos previamente llenados en la sección de interfaz del perfil que son:

- jTextFieldNombre
- jTextFieldAPaterno
- jTextFieldAMaterno

Se concatenarán de la forma, para así mostrarlos juntos dentro del jTextFieldNomComPac.

```
jTextFieldNomComPac.setText(jTextFieldNombre.getText()+jTextFieldAPaterno.getText() + jTextFieldAMaterno.getText());
```

Cuando se dé clic en el botón guardar los datos previamente tomados se almacenaron de la siguiente manera en la base de datos.

Los datos de jTextFieldNombre se insertarán en la tabla de paciente dentro del campo nombre, que está en la base de datos, con las siguientes restricciones:

- Solo debe contener letras
- Longitud máxima de 135 caracteres
- No vacío

Después para el dato contenido en jTextFieldAPaterno que corresponde al campo primer\_apellido dentro de nuestra tabla de pacientes se seleccionara de la sección perfil , toman en cuenta las mismas restricciones:

- Solo debe contener letras
- Longitud máxima de 135 caracteres

- No vacío

El siguiente dato capturado dentro de jTextFieldAPaterno que se insertara en el campo segundo\_apellido dentro de nuestra tabla de pacientes, también contendrá estas restricciones:

- Solo debe contener letras
- Longitud máxima de 135 caracteres
- No vacío

## ➤ Caso 2

En este caso , es cuando los datos contenidos en las cajas de texto de la sección de perfil si coinciden con los datos de un registro de la tabla paciente.

Entonces se realiza la siguiente acción para los 3 datos que corresponden al llenado de jTextFieldNomComPac.

```
SELECT * FROM paciente WHERE nombre
```

```
SELECT * FROM paciente WHERE primer_apellido
```

```
SELECT * FROM paciente WHERE segundo_apellido
```

Una vez que se logren encontrar los datos se concatenaran de la siguiente forma

```
SELECT nombre,primer_apellido , segundo_apellido FROM paciente
WHERE CONCAT ('nombre,',primer_apellido,',',segundo_apellido')
```

Seguido de esta concatenación se mandaran a lo que es el jTextFieldNomComPac

- Componente(s) relacionados

- JButtonGuardar
- JButtonModificar
- JButtonConsultar
- JButtonAgregar

## 54. JLabelFolio

Etiqueta que nos indicara en donde se insertara y mostrara el folio del paciente , también mostrara el mensaje de “Folio Asignado”.

## 55. jTextFieldFolio

En esta caja de texto podremos insertar , guardar y visualizar el folio del paciente. Este dato se almacenara en la base de datos cliente.

## ➤ Caso 1

Dado de que no se tenga un folio ,lo mas probable es que no haiga un registro ya que el folio asignado es un auto incrementable , mismo que sirve como ID para identificar el historial clínico de cierta persona.

Este se generara cuando el seleccionemos la opción agregar , debido a que no hay ningún registro correspondiente entre los datos de la sección perfil y datos en la tabla de paciente de la base de datos.

Y debe de obedecer las siguientes restricciones.

- Generado automáticamente (Campo de la BD?)
- No admite modificación

#### ➤ Caso 2

Si los datos en la sección de perfil y los datos contenidos en algún registro de la tabla de paciente de la base de datos.

La manera de insertar el folio en el jTextFieldFolio será:

```
jTextFieldFolio = SELECT * FROM paciente WHERE id ;
```

- Componente(s) relacionados

- jButtonConsultar
- jButtonAgregar
- jTextFieldCurp2

#### 56. jLabelCurp2

Esta etiqueta nos indica donde se insertara y mostrara el curp.

#### 57. jTextFieldCurp2

Aquí en esta caja de texto nos permitirá insertar, guardar y visualizar el curp del paciente, el dato de curp se podrá almacenara y extraerá de la base de datos de paciente.

#### ➤ Caso 1

Si el curp capturado en el jTextFieldCurp , no corresponde a ninguno de los registros capturados debido a que ninguno de estos últimos no lo contiene, debemos de plantear y hacer una serie de restricciones o formato para poder capturarlo e insertarlo en la base de datos.

- Alfanumérico únicamente
- Primeros 4 caracteres solo letras, correspondientes a las iniciales de su nombre completo
- Los siguientes 6 caracteres solo números , que corresponden a la fecha de nacimiento.

- El siguiente un solo carácter definiendo su sexo
- Los 2 siguientes, entidad federativa
- Los siguientes 3 las , las primera 3 consonantes de su apellido materno
- El penúltimo , indica el diferenciador de homonimia y siglo
- El ultimo solo indica el dígito verificador
- No vacío

➤ **Caso 2**

En este caso, si el curp corresponde a un registro dentro de la tabla paciente , se extraerá el curp de la siguiente manera

```
SELECT * FROM paciente WHERE curp
```

- Componente(s) relacionados

- jButtonGuardar
- jButtonModificar
- jButtonConsultar
- jButtonAgregar

#### 58. jLabelFecNac

Esta etiqueta nos indicara donde corresponde insertar y visualizar la fecha de nacimiento del paciente.

#### 59. jTextFieldFecNac

En esta caja de texto podemos inserta el CURP del paciente, donde una vez sea guardado se podrá visualizar en esta caja de texto.

➤ **Caso 1**

Una vez que demos clic en el botón agregar , nos habilitara la caja de texto de jTextFieldNac , en donde deberemos llenar los datos correspondientes a la fecha de nacimiento del paciente

El formato que se debe seguir es día/mes/año

Tomando en cuenta este formato los días no deben ser mayores a 31 , los meses mayores a 12 y el año de nacimiento no debe ser mayor al año en curso.

El condicionamiento y restricciones son mas o menos seria así

- Día <= 31
- Mes <=12
- Año < 2019 o año en curso
- No vacío

➤ **Caso 2**

Dado que si la fecha de nacimiento corresponde a un registro específico en base al nombre completo del paciente y su curp , se mostrara en la jTextFieldFecNac y se extraerá de la siguiente manera

```
SELECT "*" FROM paciente WHERE fecha_nacimiento.
```

- Componente(s) relacionados
  - JButtonGuardar
  - JButtonModificar
  - JButtonConsultar

#### 60.jLabelTelefono

Esta etiqueta con el mensaje de "Teléfono:" , nos indica que la caja de texto que esta a su lado derecho corresponde insertar el teléfono del paciente.

#### 61.jTextFieldTelefono

Aquí en esta caja de texto se nos permitirá insertar, guardar y visualizar el teléfono del paciente, mismo dato se podrá almacenar y extraer de la base de datos paciente.

##### ➤ Caso 1

En el caso de que no existan registros en la base datos correspondientes a un paciente, se procede a mostrar la opción a agregar y cuando se habilite el jTextFieldTelefono , debemos insertar un numero telefónico, las restricciones y formato que debe seguir es:

- Solo números
  - Dado que estamos en mexico la lada es +52, esta se da por defecto una vez capturado el teléfono
  - Los siguientes serán 10 números, si en los 10 espacios solo se insertan 0 (ceros) , mandaremos una alerta de incorrecto.
  - El numero en si concatenado quedara num = +52 +"  
jTextFieldTelefono.getText()";
  - No vacío
- **Caso 2**

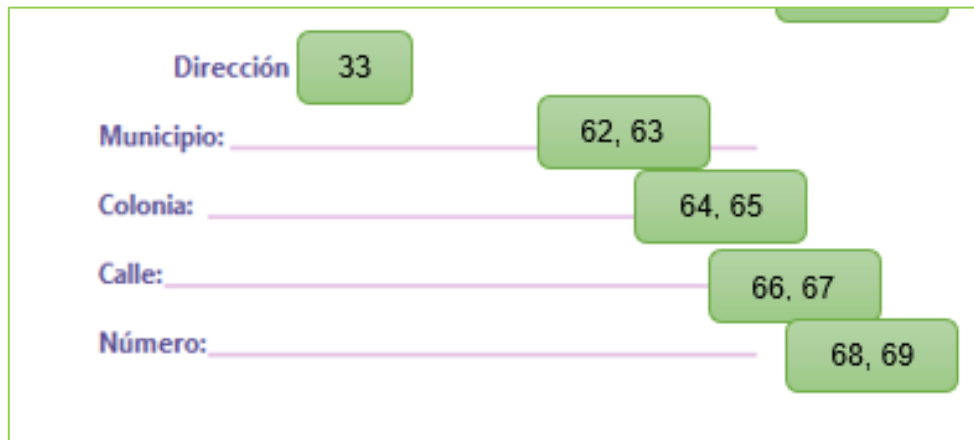
Si existen datos contenidos en la tabla paciente y el teléfono , lo mostraremos en en el jextFieldTelefono cuando demos clic en el botón consultar

La manera de jalarlo será

```
SELECT * FROM paciente WHERE teléfono.
```

- Componente(s) relacionados

- JButtonGuardar
- JButtonModificar
- JButtonConsultar



33	jLabelTitulo4(Dirección)
62	jLabelMunicipio
63	jTextFieldMunicipio
64	jLabelColonia
65	jTextFieldColonia
66	jLabelCalle
67	jTextFieldCalle
68	jLabelNumero
69	jTextFieldNumero

### 32.jLabelTitulo4(Dirección)

Esta etiqueta mostrara el mensaje diciendo “Dirección” , misma que nos indicara el aparatado de la dirección del cliente.

### 63.jLabelMunicipio

Esta etiqueta nos muestra el mensaje “Municipio” , mismo que hace referencia de donde colocar el municipio en el que vive el paciente.



#### 64. jTextFieldMunicipio

Es una caja de texto que nos permitirá insertar, guardar y visualizar el municipio en el que reside el paciente.

- **Caso 1**

Si no existe ningún registro en base a datos del paciente capturados en perfil relacionados con un registro de la base de datos , entonces se seleccionara la opción de agregar.

En si el jTextFieldMunicipio , recibirá

- Longitud máxima de 45 caracteres
- Dato de tipo cadena
- No vacío

- **Caso 2**

Dado que si hay datos relacionados en la base de datos , entonces se tomara el dato del campo municipio de la tabla paciente en la base de datos , de esta manera.

```
SELECT * FROM paciente WHERE municipio
```

- Componente(s) relacionados

- jButtonGuardar
- jButtonModificar
- jButtonConsultar

#### 65. jLabelColonia

Esta etiqueta hace alusión en donde se insertara y se mostrara la colonia de la dirección del paciente

#### 66. jTextFieldColonia

Una caja de texto que nos servirá como medio de captura y visualización de la colonia en la que se ubica la dirección del paciente.

- **Caso 1**

Si no existe ningún registro en base a datos del paciente capturados en perfil relacionados con un registro de la base de datos , entonces se seleccionara la opción de agregar. Lo que se capture en jTextFieldColonia corresponderá a la colonia en la que reside el paciente, este dato debe seguir la siguientes restricciones.

- Longitud maxima de 45 caracteres
- Solo debe contener letras

- No vacío

➤ **Caso 2**

Ahora si es que si hay algún dato registrado en la base de datos con relación a los datos capturados en perfil , procedemos a extraer el dato de colonia de la siguiente manera.

```
SELECT * FROM paciente WHERE colonia
```

- Componente(s) relacionados

- jButtonGuardar
- jButtonModificar
- jButtonConsultar

#### 66.jLabelCalle

Eiqueta que nos muestra un mensaje “Calle” , mismo que indica donde insertar y visualizar la calle de la dirección del paciente.

#### 67.jTextFieldCalle

En esta caja de texto se nos permitirá insertar y una vez guardada la información se mostrara lo que es la calle en la que se encuentra la dirección del paciente

➤ **Caso 1**

Si no se encontraron datos relacionados con respecto al perfil y a la tabla paciente dentro de base de datos , se da clic en agregar , se nos habilita el jTextFieldCalle , el dato a almacenar contendrá estas restricciones.

- Longitud maxima de 45 caracteres
- Dato de tipo cadena
- No vacío

➤ **Caso 2**

Ahora si hay datos que concuerdan entre el perfil y la table de paciente , solo se extraera el dato de calle , para mostrarlo en el jTextFieldCalle, la forma de traerlo es así

```
SELECT * FROM paciente WHERE calle
```

- Componente(s) relacionados

- jButtonGuardar
- jButtonModificar
- jButtonConsultar.

#### 68. JLabelNumero

Esta etiqueta nos indicara donde se debe insertar y visualizar el numero con respecto a la dirección del paciente.

#### 69. JTextFieldNumero

Es una caja de texto que nos permite guardar y visualizar el numero correspondiente a la dirección del paciente.

##### ➤ Caso 1

Una vez que hayamos clicado en agregar , el JTextFieldNumero se habilitara y deberemos insertar el dato que debe seguir esta restriccion

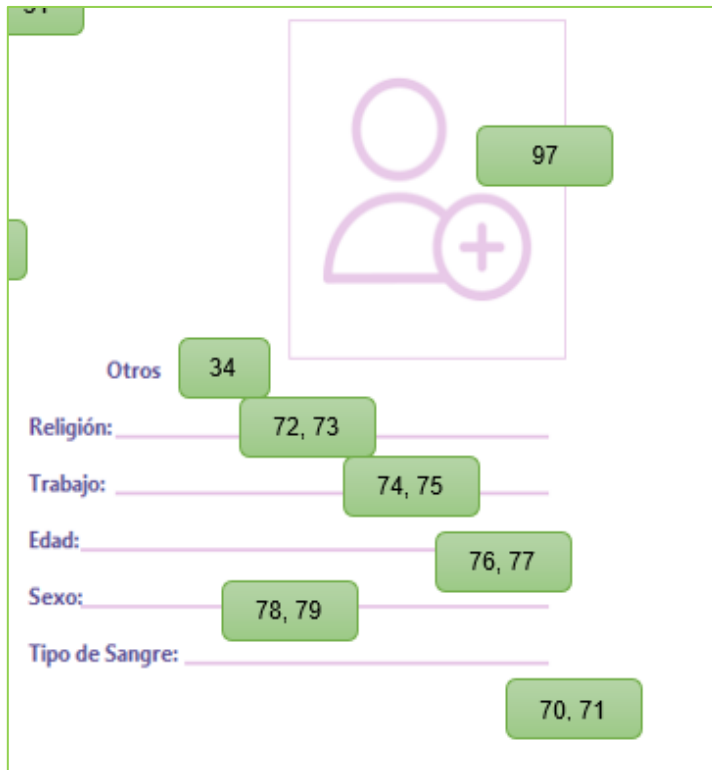
- Dato de tipo numérico
- No vacío

##### ➤ Caso 2

Si hay datos que corresponden a cierto registro el JTextFieldNumero nos retornara un dato , que se extrae desde la base de datos de la siguiente manera

```
SELECT * FROM paciente WHERE numero
```

- Componente(s) relacionados
  - JButtonGuardar
  - JButtonModificar
  - JButtonConsultar



97	jButtonInsertImg
34	jLabelTitulo5(Otros)
72	jLabelReligion
73	jTextFieñdReligion
74	jLabelTrabajo
75	jTextFieldTrabajo
76	jLabelEdad
77	jTextFieldEdad
78	jLabelSexo
79	jTextFieldSexo
70	jLabelTipSangre
71	jTextFieldTipSangre

#### 97.jButtonInsertImg

##### ➤ caso 1

Quando ciclemos la opción agregar , este jButtonInsertigng que al momento de dar clic nos permite agregar una foto del paciente.

- Solo imágenes(xxx.png , xxx.jpg, xxx.jpeg)

➤ **Caso 2**

Si ya existen registros relacionados con datos capturados en perfil con datos almacenados en tabla de paciente , se nos descargara la imagen en ese JButtonInsering.

- Componente(s) relacionados

- JButtonGuardar
- JButtonModificar
- JButtonConsultar

34. JLabelTitulo5(Otros)

Esta etiqueta nos muestra un mensaje diciendo “Otros” , esto indica un apartado diferente con los datos adicionales del paciente.

72. JLabelReligion

Esta equeta nos indica donde se insertara y visualizara la religión del paciente.

73. jTextFieñdReligion

Es un caja de texto que nos permitirá guardar y ver la religión que profesa el paciente.

➤ **Caso 1**

Una vez que ciclemos en agregar , el jTextFieldReligion se habilitara y podremos insertar el dato que tiene que seguir estas restricciones

- Longitud maxima de 20 caracteres
- Solo debe contener letras
- No vacío, si es vacío que nos muestre al momento de consultar “Ninguna”

➤ **Caso 2**

Si hay registros en la tabla paciente que sean los mismos de perfil , el jTextFieldReligion se llenara con la, religión que profese el paciente , si no profesa ninguna , en ese campo nos marcara “Ninguna.”

Se seleccionará de esta manera

```
SELECT * FROM paciente WHERE religion
```

- Componente(s) relacionados

- jButtonGuardar
- jButtonModificar
- jButtonConsultar

#### 74. jLabelTrabajo

En esta etiqueta se nos muestra el mensaje de “Trabajo”, este indica donde se puede insertar o ver el trabajo del paciente.

#### 75. jTextFieldTrabajo

Aquí en esta caja de texto se permitirá guardar y mostrar el trabajo que desempeña el paciente.

##### ➤ Caso 1

Si ningún registro concuerda con lo capturado en perfil , debemos dar clic en agregar y se nos habilita jTextFieldTrabajo , el dato que vamos a contener aquí tiene las siguientes restricciones.

- Longitud máxima de 50 caracteres
- Solo debe contener letras
- No vacío

##### ➤ Caso 2

Si existen dato relacionados entre el perfil y un registro en la tabla paciente, en el jTextFieldTrabajo se nos mostrara un dato con respecto a la labor que realiza el paciente.

La extraeremos desde la base de datos de la siguiente manera

```
SELECT * FROM paciente WHERE trabajo
```

- Componente(s) relacionados

- jButtonGuardar
- jButtonModificar
- jButtonConsultar

#### 76. jLabelEdad

Esta etiqueta nos mostrara un texto diciendo “Edad:” mismo que indicara donde se insertara y visualizara la edad del paciente.

#### 77. jTextFieldEdad

Una caja de texto que solo admite números y nos permite guardar o visualizar la edad del paciente.

##### ➤ Caso 1

En caso de que no coincidan datos del perfil con datos en los registros de la tabla paciente , debemos dar clic en la opción agregar , y en el jTextFieldEdad , el dato edad que seguirá estas restricciones.

- Dato de tipo numérico
- No vacío
- No admite modificación

➤ **Caso 2**

Si los datos en perfil coinciden con algún registro de la tabla paciente el jTextFieldEdad se llenará como consulta, extrayendo datos de la tabla paciente de la siguiente forma

```
SELECT * FROM paciente WHERE edad
```

- Componente(s) relacionados
  - JButtonConsultar
  - JTextFieldFecNac

#### 78. jLabelSexo

Es una etiqueta con el texto “Sexo:” que hace referencia de donde guardar y visualizar el sexo del paciente.

#### 79. jTextFieldSexo

Es una caja de texto que solo admitirá capturar H si es hombre y M Si es mujer , de acuerdo con los datos de paciente.

➤ **Caso 1**

Si no existen datos que sean similares a los capturados a los de perfil , emergerá la ventana de error y la opción de agregar , cuando demos agregar el jTextFieldSexo se habilitara y el dato a insertar seguirá estas restricciones.

- Dato de tipo carácter
- H / M (Tamaño 1)
- No vacío

➤ **Caso 2**

En el caso de que si existan los datos insertados en perfil con algún registro de la tabla paciente , el jTextFieldSexo se llenara de la siguiente manera ,con el dato correspondiente al campo sexo de la tabla paciente

```
SELECT * FROM paciente WHERE sexo
```

- Componente(s) relacionados
  - JButtonGuardar
  - JButtonModificar
  - JButtonConsultar

#### 70. JLabelTipSangre

Es una etiqueta que nos mostrara un mensaje diciendo “Tipo de sangre:”,referenciando donde insertar y visualizar el tipo de sangre que porta el paciente.

#### 71. JTextFieldTipSangre

Caja de texto que permitirá capturar datos de tipo cadena , para guardar y visualizar el tipo de sangre del paciente.

##### ➤ Caso 1

Si los datos capturados en perfil no corresponden a algún registro de la tabla paciente en la base de datos , emergerá un mensaje de error y la opción agregar, se nos habilitara el jTextFieldSangre y debemos llenarlo en base a estas restricciones.

- Longitud de 5
- Coincidir con el patrón dd/+-
- No vacío

##### ➤ Caso 2

Si los datos de perfil coinciden con algún registro de la tabla paciente de la base de datos , ahora solo extraeremos este dato para mostrarlo en el jTextFieldTipSangre.

De la siguiente forma

```
SELECT * FROM paciente WHERE tipo_sangre
```

- Componente(s) relacionados
  - JButtonGuardar
  - JButtonModificar
  - JButtonConsultar



Obtención de la información

```
SELECT
    img,
    fecha_nacimiento,
    sexo,
    tipo_sangre,
    religion,
    telefono,
    trabajo,
    calle,
    numero,
    colonia,
    municipio,
    CONCAT(first_name, ' ', last_name) full_name
WHERE
    curp = '$CURP'
FROM
    pacientes;
```

```
try {
    ResultSet rs = DB.executeResultSetQuery(
        "select xxxxxxxx " +
        "from pacientes where CURP = " + jTextFieldCurp.getText());
    while (rs.next()) {
        String id = rs.getString("id");
        String nombre = rs.getString("nombre");
        String sexo = rs.getString("sexo");
        .....
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```

Validación de cajas de texto:

```
private static boolean alphabetic(JFXTextField node) {
    Pattern p = Pattern.compile("[ a-zA-ZñÑáéíóúÁÉÍÓÚ]+");
    return isMatch(node, p);
}
```

Validación tipo sangre:

```
private static boolean sangre(JFXTextField node) {  
    Pattern p = Pattern.compile("/(A|B|AB|O)[+-]/");  
    return isMatch(node p);  
}
```

Validación de fecha ejemplo:

```
public static void date(JFXDatePicker datePicker){  
    if (datePicker.getValue() != null) {  
        if (datePicker.getValue().compareTo(LocalDate.now()) != -1) {  
            LocalDate date = datePicker.getValue();  
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern("EEEE d 'de' MMMM 'de' yyyy");  
            String formattedString = date.format(formatter);  
            System.out.println(formattedString);  
        } else {  
            Interface.clear(datePicker);  
            Messages.simpleMessage(datePicker, "Error", "Please, enter a valid date");  
        }  
    }  
}
```

Validación teléfono:

```
private static boolean phone(JFXTextField node) {  
    Pattern p = Pattern.compile("[/^[\+]?([0-9]{3})?[-\s\.]?[0-9]{3}[-\s\.]?[0-9]{4,6}$/im]+");  
    return isMatch(node, p);  
}
```