

Entrega 4 del proyecto de Programación Orientada a Objetos

Fecha de entrega: 05/11/2020

Jun Woo Lee Hong 20358

Cristian Eduardo Aguirre Duarte 20231

Andrés Alejandro de la Roca Pineda 20332

Sayra Estefanía Elvira Ramos 20725

Pablo Daniel Gonzalez Ramos 20362

Manuel Alejandro Archila Moran 161250

Profundización sobre requisitos funcionales

- **Funcion de Login de usuario:** Para esta entrega se creó una interfaz gráfica que maneja todas las funcionalidades del ingreso del usuario, se creó una interfaz para el login y otro para el registro de usuarios, cumpliendo con las sugerencias de retroalimentación de algunos usuarios de volver la interfaz en una más amigable.
- **Cálculo de riesgo del paciente frente a la enfermedad:** La interfaz para el cálculo de riesgo se realizó en forma gráfica con el objetivo de presentar la información al usuario de una manera más amigable.
- **Preguntarle al usuario sobre sus síntomas:** A este requisito se le agregó una interfaz en la que al igual que en la entrega anterior se le preguntará al usuario sobre sus síntomas para posteriormente sugerir que pruebe algún tratamiento.
- **Información sobre el COVID-19:** Para este requisito solamente se profundizó la interfaz para volverla gráfica, de esta manera le podemos mostrar a los usuarios información sobre el virus de una forma amigable y concisa, más optimizada que leerla directamente de una línea en la terminal.
- **Contacto a números de emergencia:** Para el requisito de contactos de emergencia se trabajó en una interfaz gráfica, ahora los números de emergencia se muestran dentro de una ventana.
- **Páginas con información pertinente sobre el COVID-19:** Para esta funcionalidad además de haberle agregado una interfaz se le agregaron hipervínculos a las páginas para volver la interacción más dinámica.

Identificación y descripción de clases necesarias

- **Clase PreCovidDriver**
 - Descripción de clase:
 - Esta clase sirve como el main, llama a la clase Login para realizar el programa.
- **Clase Login**

- Descripción de clase:
 - Esta clase cumplirá con la función de dar la opción a los usuarios de crear nuevas cuentas o ingresar con cuentas ya registradas previamente. También al crear una nueva cuenta, les pide al usuario información como nombre, vivienda, edad, correo electrónico y su contraseña.
- Atributos
 - javax.swing.JPasswordField inputPassword
 - Este atributo es para que el usuario ingrese su contraseña
 - javax.swing.JTextField inputUsername
 - Este atributo es para que el usuario ingrese su usuario
 - javax.swing.JButton jButtonSignIn
 - Este atributo es para que el usuario ingrese al programa
 - javax.swing.JButton jButtonSignUp
 - Este atributo es para que el usuario cree una cuenta nueva.
 - javax.swing.JLabel jLabelClose
 - Este atributo es para cerrar el programa.
 - javax.swing.JLabel logo
 - Este atributo es para mostrar el logo del programa
 - javax.swing.JLabel userLogo
 - Este atributo es para mostrar una imagen
 - javax.swing.JLabel lockLogo
 - Este atributo es para mostrar una imagen
 - javax.swing.JPanel bgColor1
 - Este atributo es para poner un color de fondo
 - javax.swing.JPanel bgColor2
 - Este atributo es para poner un color de fondo
- Métodos
 - *Login*
 - Llama al método initComponents
 - *initComponents*
 - Este método es para crear la interfaz del login con todos los botones y cajas de texto.
 - *inputUsernameActionPerformed(java.awt.event.ActionEvent evt)*
 - Este método es para guardar lo que se escribe en la caja de usuario.
 - *inputPasswordActionPerformed(java.awt.event.ActionEvent evt)*
 - Este método es para guardar lo que se escribe en la caja de contraseña
 - *jButtonSignInActionPerformed(java.awt.event.ActionEvent evt)*

- Este método es para lo que pasa cuando el usuario hace click en el botón de sign in guarda lo que el usuario ingreso en las líneas de texto y las compara con el usuario y contraseña guardada en el csv.
 - *jButtonSignUpActionPerformed(java.awt.event.ActionEvent evt)*
 - Este método es para cerrar la página de sign in y abrir la del sign up.
 - *jLabelCloseMouseClicked(java.awt.event.MouseEvent evt)*
 - Este método es para cerrar el programa.
- **Clase CreateAccount**
 - Descripción de clase:
 - Esta clase cumplira de dar la opcion a los usuarios para crear nuevas cuentas dándoles los espacios para ingresar sus datos y luego guardar los datos en un csv.
 - Atributos:
 - `javax.swing.JPanel bgColor1;`
 - Este atributo es para poner un color de fondo
 - `javax.swing.JPanel bgColor2;`
 - Este atributo es para poner un color de fondo
 - `javax.swing.JButton jButtonCreate;`
 - Este atributo es para crear las cuentas y guardarlas al csv
 - `javax.swing.JButton jButtonRegresar;`
 - Este atributo es para regresar al log in
 - `javax.swing.JLabel jLabelAddress;`
 - Este atributo es para mostrar un texto de la vivienda
 - `javax.swing.JLabel jLabelAge;`
 - Este atributo es para mostrar un texto de la edad
 - `javax.swing.JLabel jLabelClose;`
 - Este atributo es para cerrar el programa
 - `javax.swing.JLabel jLabelEmail;`
 - Este atributo es para mostrar un texto de correo
 - `javax.swing.JLabel jLabelName;`
 - Este atributo es para mostrar un texto de nombre
 - `javax.swing.JLabel jLabelPassword;`
 - Este atributo es para mostrar un texto de contraseña
 - `javax.swing.JLabel jLabelUsername;`
 - Este atributo es para mostrar un texto de usuario
 - `javax.swing.JLabel text;`
 - Este atributo es para mostrar un texto de título
 - `javax.swing.JTextField textAddress;`
 - Este atributo es para que el usuario ingrese su vivienda
 - `javax.swing.JTextField textAge;`
 - Este atributo es para que el usuario ingrese su edad

- `javax.swing.JTextField textEmail;`
 - Este atributo es para que el usuario ingrese su correo
- `javax.swing.JTextField textName;`
 - Este atributo es para que el usuario ingrese su nombre
- `javax.swing.JTextField textPassword;`
 - Este atributo es para que el usuario ingrese su contraseña
- `javax.swing.JTextField textUsername;`
 - Este atributo es para que el usuario ingrese su usuario.

○ Métodos

- `initComponents()`
 - Este método es para crear la interfaz para crear cuentas con todos los botones y cajas de texto.
- `jLabelCloseMouseClicked(java.awt.event.MouseEvent evt)`
 - Este método es para cerrar el programa
- `jButtonCreateActionPerformed(java.awt.event.ActionEvent evt)`
 - Este método es para tomar toda la información que el usuario ingresó en las cajas y guardarlas en un csv.
- `jButtonRegresarActionPerformed(java.awt.event.ActionEvent evt)`
 - Este método es para regresar al log in.

● **Clase ContactosEmergencia**

○ Descripción de clase:

- Esta clase cumplirá con el requisito funcional de contactos a números de emergencia, se encargará de proporcionarle al usuario estos números telefónicos, dentro de la interfaz podemos encontrar números ya sea de emergencias públicas o privadas.

○ Atributos:

- `javax.swing.JTextArea ContactosPrivados;`
 - Este es un atributo que se encarga de mostrar un recuadro de texto en donde están escritos los contactos de emergencia privados.
- `javax.swing.JTextArea ContactosPublicos;`
 - Este atributo se encarga de mostrar un recuadro de texto en donde están escritos los contactos de emergencia públicos.
- `javax.swing.JButton ExitButton;`

- Este atributo es un botón que se encarga de salir de la ventana que contiene los contactos de emergencia para regresar al menú principal del programa.
 - javax.swing.JLabel LabelTitulo:
 - Este atributo es una etiqueta que se encarga de mostrar de la ventana dentro de la interfaz.
 - javax.swing.JLabel jLabel2:
 - Este atributo se encarga de otorgarle una imagen de fondo a la ventana.
 - javax.swing.JScrollPane jScrollPane1:
 - Este atributo se encarga de permitir que el usuario se mueva a través del JTextArea de ContactosPrivados.
 - private javax.swing.JScrollPane jScrollPane2:
 - Este atributo se encarga de permitir que el usuario se mueva a través del JTextArea de ContactosPublicos
- Métodos
 - initComponents():
 - Este método se encarga de iniciar todos los objetos que forman parte de la interfaz gráfica.
 - ExitButtonActionPerformed(java.awt.event.ActionEvent evt):
 - Este método se encarga de la funcionalidad del botón de salida del programa, permite regresar al menú principal y salir de la ventana de ContactosEmergencia.
- **Clase OperacioneS(síntomas):**
 - Descripción de la clase:
 - Esta clase cumplirá con el requisito funcional de solicitarle al usuario que ingrese los síntomas que él está padeciendo, así mismo le brindará ciertos remedios y cuidados caseros que deberá tener para ser atendido en casa, de igual forma se brinda una serie de medicamentos que están establecidos para todas las personas que quieran tratar al coronavirus en su hogar.
 - Atributos:
 - String Array SintomasComunes:
 - este array almacena información de todos los síntomas más comunes que presenta el coronavirus.
 - String Array SintomasMenosComunes:
 - este array almacena información de todos los síntomas que son tan inusuales en el virus del coronavirus.

- String Array SintomasGraves:
 - Este array almacena información de los síntomas más graves, es decir todos aquellos que ya no pueden ser tratados en la comodidad del hogar, sino que deben ser asistidos por personal médico.
 - Métodos :
 - IngresarSintomas:
 - Se encarga de solicitarle al usuario que ingrese los síntomas que está padeciendo, este los clasifica en las categorías de los más comunes, los menos inusuales y los más graves, para así determinar los tips y medicamentos que recomendará según los síntomas que el individuo presenta.
- **Clase HistorialMedico:**
 - Esta clase se encarga de solicitar informacion al paciente sobre su historial médico, es decir padecimientos que pueda tener y que son factor de riesgo para el Covid, además de ello solicita informacion como la edad, sexo y en base a ello realiza un calculo de probabilidad de riesgo.
 -
 - Atributos:
 - **GrupoDeBotones = new javax.swing.ButtonGroup();** Este atributo agrupa los botones de masculino y femenino para que solo una opción pueda ser seleccionada.
 - **Masculino = new javax.swing.JRadioButton();** Boton de sexo masculino perteneciente al grupo de botones de sexo.
 - **Femenino = new javax.swing.JRadioButton();** Boton de sexo femenino perteneciente al grupo de botones de sexo.
 - **BotonesDeEdad = new javax.swing.ButtonGroup();** Este atributo agrupa los botones de rango de edades para que solo una opción pueda ser seleccionada.
 - **E1 = new javax.swing.JRadioButton();** Boton de edad de 0 a 20 años perteneciente al grupo de botones de sexo.
 - **E2 = new javax.swing.JRadioButton();** Boton de edad de 20 a 30 años perteneciente al grupo de botones de sexo.
 - **E3 = new javax.swing.JRadioButton();** Boton de edad de 30 a 40 años perteneciente al grupo de botones de sexo.
 - **E4 = new javax.swing.JRadioButton();** Boton de edad de 40 a 60 años perteneciente al grupo de botones de sexo.
 - **E6 = new javax.swing.JRadioButton();** Boton de edad de 60 a 80 años perteneciente al grupo de botones de sexo.
 - **E5 = new javax.swing.JRadioButton();** Boton de edad de 80 a 100 años perteneciente al grupo de botones de sexo.

- **jLabel1 = new javax.swing.JLabel();** Atributo que funciona como una tiqueta de texto para indicar cada apartado
- **jLabel2 = new javax.swing.JLabel();** Atributo que funciona como una etiqueta de texto para indicar cada apartado
- **jLabel3 = new javax.swing.JLabel();** Atributo que funciona como una etiqueta de texto para indicar cada apartado
- **jLabel4 = new javax.swing.JLabel();** Atributo que funciona como una etiqueta de texto para indicar cada apartado
- **jLabel5 = new javax.swing.JLabel();** Atributo que funciona como una etiqueta de texto para indicar cada apartado
- **jLabel6 = new javax.swing.JLabel();** Atributo que funciona como una etiqueta de texto para indicar cada apartado
- **B2 = new javax.swing.JCheckBox();** Cuadro de verificación para indicar si el paciente padece o no la enfermedad.
- **B3 = new javax.swing.JCheckBox();** Cuadro de verificación para indicar si el paciente padece o no la enfermedad.
- **B4 = new javax.swing.JCheckBox();** Cuadro de verificación para indicar si el paciente padece o no la enfermedad.
- **B1 = new javax.swing.JCheckBox();** Cuadro de verificación para indicar si el paciente padece o no la enfermedad.
- **jButton1 = new javax.swing.JButton();** Botón utilizado para realizar el cálculo de probabilidades
- **jButton2 = new javax.swing.JButton();** Botón utilizado para cerrar la ventana

○ Métodos:

- **HistorialMedico()** Metodo constructor en el cual se agrupan los botones individuales dentro de un grupo de botones, de este modo, solo se puede seleccionar uno de la categoría que corresponda.
- **initComponents()** Este método se encarga de iniciar todos los objetos que forman parte de la interfaz gráfica.
- **jButton2ActionPerformed(java.awt.event.ActionEvent evt)** Este método realiza todos los cálculos de probabilidad una vez el usuario haya seleccionado todas las casillas y haya presionado el botón de calcular.
- **jButton1ActionPerformed(java.awt.event.ActionEvent evt)** Este metodo ejecuta la instrucción de cerrar ventana cuando el usuario hace clic sobre el.

● **Clase InfoCovid:**

○ Descripción de la clase:

- Esta clase cumplirá con proveer al usuario con la información básica necesaria para entender la seriedad de la enfermedad. Esta clase brinda

al usuario información sobre los posibles síntomas que este virus puede causar, al igual que brindar los posibles cuidados y precauciones para no contagiarse. Por último, presenta la información de las maneras en las que una persona se puede infectar.

- Atributos:

-

- Métodos:

- DesplegarManeras:

- Este método devuelve un String con las maneras en que el virus puede ser contraído por una persona, al igual que señalar lo fácil que es contagiarse de este virus.

- **Clase Display1:**

- Descripción de clase: Esta clase es la interfaz gráfica para mostrar los síntomas comunes y serios para reconocer si se tiene COVID-19. También muestra algunas maneras en las que alguien se puede contagiar de COVID-19. Esta tiene botones que llevan a referencias externas que son páginas de la OMS

- Atributos

- javax.swing.JButton jButton1 : Botón que dirige a la información de la página de la OMS
- javax.swing.JButton jButton2 : Botón que redirige a la página de la OMS sobre rumores esparcidos del virus
- javax.swing.JButton jButton3 : Botón que redirige a la página de la OMS acerca de viajes seguros.
- javax.swing.JLabel JLabel1 : Título de Síntomas comunes
- javax.swing.JLabel JLabel2 : Título de síntomas serios
- javax.swing.JLabel JLabel3 : Título para las maneras de contraer COVID-19
- javax.swing.JLabel JLabel4: Título para los cuidados de COVID-19
- javax.swing.JLabel JLabel5 : Título para más información
- javax.swing.JList<String> JList1 : Lista con los síntomas comunes
- javax.swing.JList<String> JList2 : Lista con los síntomas serios
- javax.swing.JList<String> JList3 : Lista con las maneras de protegerse
- javax.swing.JTextArea JResul : Area de texto donde van los maneras en las que alguien se puede enfermar
- javax.swing.JScrollPane jScrollPane1 : Panel que proporciona scroll para ver todo el texto
- javax.swing.JScrollPane jScrollPane2 : Panel que proporciona scroll para ver todo el texto
- javax.swing.JScrollPane jScrollPane3 : Panel que proporciona scroll para ver todo el texto

- javax.swing.JScrollPane jscrollPane4 : Panel que proporciona scroll para ver todo el texto
 - javax.swing.JScrollPane jscrollPane5: Panel que proporciona scroll para ver todo el texto
- Métodos
 - initComponents(): Método que inicializa los componente de la interfaz gráfica
 - Display(): Método constructor de la interfaz.
 - jButton1ActionPerformed(java.awt.event.ActionEvent evt) : Método que lleva a la página de la OMS para más información cuando se hace click sobre el.
 - jButton2ActionPerformed(java.awt.event.ActionEvent evt) : Método que lleva a la página de rumores del COVID-19 al hacer click sobre él.
 - jButton3ActionPerformed(java.awt.event.ActionEvent evt) : Método que lleva a la página de recomendaciones para viajes seguros al hacer click sobre él.
 -
- **Clase DirectorioHospitales:**
 - Descripción de la clase:
 - Esta clase cumplirá con el requisito funcional de poder mostrar al usuario una base de datos de algunos de los Hospitales que existen dentro del perímetro Guatemalteco donde se realizan pruebas de Covid19. Estos se mostrarán organizados por departamento, el nombre del hospital y el número de teléfono del Hospital. El número de teléfono del hospital es algo muy importante debido a que el usuario podrá realizar la llamada al hospital que más le convenga para luego corroborar que existan pruebas disponibles en este hospital para luego ir a realizarse la prueba y si no hay buscar otro hospital en la base de datos.
 - Atributos:
 - ArrayList hospitales: Este atributo es uno de los más importantes debido a que en este se almacenarán los 22 HashMaps de cada departamento y dentro de cada Hasap de departamento habrá Hasmap de los hospitales que están dentro de este departamento con su nombre y su numero de telefono.
 - ArrayList Guatemala: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Guatemala.
 - ArrayList Sanmarcos: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de San Marcos.
 - ArrayList Quetzaltenango: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Quetzaltenango.
 - ArrayList Huehuetenango: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Huehuetenango.

- Arraylist Quiché: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Quiché
- Arraylist AltaVerapaz: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Alta Verapaz.
- Arraylist BajaVerapaz: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Baja Verapaz.
- Arraylist ElProgreso: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de El progreso.
- Arraylist Zacapa: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Zacapa.
- Arraylist Izabal: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Izabal.
- Arraylist Petén: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Petén.
- Arraylist Chiquimula: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Chiquimula.
- Arraylist Jutiapa: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Jutiapa.
- Arraylist Jalapa: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Jalapa.
- Arraylist SantaRosa: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Santa Rosa.
- Arraylist Escuintla: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Escuintla.
- Arraylist Sacatepequez: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Baja Verapaz.
- Arraylist Retalhuleu: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Retalhuleu.
- Arraylist Sololá: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Sololá.
- Arraylist Totonicapán: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Totonicapán.
- Arraylist Chimaltenango: Este almacenará todos los Hashmaps de los hospitales que estén dentro del departamento de Chimaltenango.
- Hashmap centromedico7: Este almacenará el nombre del hospital 1 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap centromedico10: Este almacenará el nombre del hospital 2 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap centromedico12: Este almacenará el nombre del hospital 3 del departamento de Guatemala conjuntamente a su numero de telefono.

- Hashmap centromedicocaya: Este almacenará el nombre del hospital 4 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap centromediCO17: Este almacenará el nombre del hospital 5 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap centromedicocarretera: Este almacenará el nombre del hospital 6 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap centromedicoroosvelt: Este almacenará el nombre del hospital 7 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap centromedicosan: Este almacenará el nombre del hospital 8 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap centromedicovilanueva: Este almacenará el nombre del hospital 9 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap tecniz9: Este almacenará el nombre del hospital 10 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap tecnizmultime: Este almacenará el nombre del hospital 11 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmaph Herrerallerandi: Este almacenará el nombre del hospital 12 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap pilar: Este almacenará el nombre del hospital 13 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap lapaz: Este almacenará el nombre del hospital 14 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap sanjuandiosh: Este almacenará el nombre del hospital 15 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap roosvelt: Este almacenará el nombre del hospital 16 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap iggz9: Este almacenará el nombre del hospital 17 del departamento de Guatemala conjuntamente a su numero de telefono.
- Hashmap nsanmarcos: Este almacenará el nombre del hospital 1 del departamento de San Marcos conjuntamente a su numero de telefono.
- Hashmap labsanmarcos: Este almacenará el nombre del hospital 2 del departamento de San Marcos conjuntamente a su numero de telefono.
- Hashmap centromedicoquetzal: Este almacenará el nombre del hospital 1 del departamento de Quetzaltenango conjuntamente a su numero de telefono.

- Hashmap hospitalpvquet: Este almacenará el nombre del hospital 2 del departamento de Quetzaltenango conjuntamente a su numero de telefono.
- Hashmap hospitalregq: Este almacenará el nombre del hospital 3 del departamento de Quetzaltenango conjuntamente a su numero de telefono.
- Hashmap hospitalxela: Este almacenará el nombre del hospital 4 del departamento de Quetzaltenango conjuntamente a su numero de telefono.
- Hashmap igxela: Este almacenará el nombre del hospital 5 del departamento de Quetzaltenango conjuntamente a su numero de telefono.
- Hashmap hospitalnacionalhuegue: Este almacenará el nombre del hospital 1 del departamento de Quetzaltenango conjuntamente a su numero de telefono.
- Hashmap pvhuehue: Este almacenará el nombre del hospital 2 del departamento de Quetzaltenango conjuntamente a su numero de telefono.
- Hashmapa altuve: Este almacenará el nombre del hospital 3 del departamento de Quetzaltenango conjuntamente a su numero de telefono.
- Hashmap hspnquiche: Este almacenará el nombre del hospital 1 del departamento de Quiché conjuntamente a su numero de telefono.
- Hashmap naza: Este almacenará el nombre del hospital 2 del departamento de Quiché conjuntamente a su numero de telefono.
- Hashmap iggsquiche: Este almacenará el nombre del hospital 3 del departamento de Quiché conjuntamente a su numero de telefono.
- Hashmap hospitalnalta: Este almacenará el nombre del hospital 1 del departamento de Alta Verapaz conjuntamente a su numero de telefono.
- Hashmap hospitalrgc: Este almacenará el nombre del hospital 2 del departamento de Alta Verapaz conjuntamente a su numero de telefono.
- Hashmap hpvp: Este almacenará el nombre del hospital 3 del departamento de Alta Verapaz conjuntamente a su numero de telefono.
- Hashmap hop1: Este almacenará el nombre del hospital 1 del departamento de Baja Verapaz conjuntamente a su numero de telefono.
- Hashmap hop2: Este almacenará el nombre del hospital 2 del departamento de Baja Verapaz conjuntamente a su numero de telefono.
- Hashmap hop3: Este almacenará el nombre del hospital 3 del departamento de Baja Verapaz conjuntamente a su numero de telefono.
- Hashmap elp1: Este almacenará el nombre del hospital 1 del departamento de El progreso conjuntamente a su numero de telefono.
- Hashmap elp2: Este almacenará el nombre del hospital 2 del departamento de Baja Verapaz conjuntamente a su numero de telefono.

- Hashmap z1: Este almacenará el nombre del hospital 1 del departamento de Zacapa conjuntamente a su numero de telefono.
- Hashmap z2: Este almacenará el nombre del hospital 2 del departamento de Zacapa conjuntamente a su numero de telefono.
- Hashmap z3: Este almacenará el nombre del hospital 3 del departamento de Zacapa conjuntamente a su numero de telefono.
- Hashmap z4: Este almacenará el nombre del hospital 4 del departamento de Zacapa conjuntamente a su numero de telefono.
- Hashmap iz1: Este almacenará el nombre del hospital 1 del departamento de Izabal conjuntamente a su numero de telefono.
- Hashmap iz2: Este almacenará el nombre del hospital 2 del departamento de Izabal conjuntamente a su numero de telefono.
- Hashmap iz3: Este almacenará el nombre del hospital 3 del departamento de Izabal conjuntamente a su numero de telefono.
- Hashmap pt1: Este almacenará el nombre del hospital 1 del departamento de Petén conjuntamente a su numero de telefono.
- Hashmap pt2: Este almacenará el nombre del hospital 2 del departamento de Petén conjuntamente a su numero de telefono.
- Hashmap pt3: Este almacenará el nombre del hospital 3 del departamento de Petén conjuntamente a su numero de telefono.
- Hashmap chq1: Este almacenará el nombre del hospital 1 del departamento de Chiquimula conjuntamente a su numero de telefono.
- Hashmap chq2: Este almacenará el nombre del hospital 2 del departamento de Chiquimula conjuntamente a su numero de telefono.
- Hashmap chq3: Este almacenará el nombre del hospital 3 del departamento de Chiquimula conjuntamente a su numero de telefono.
- Hashmap jt1: Este almacenará el nombre del hospital 1 del departamento de Jutiapa conjuntamente a su numero de telefono.
- Hashmap jt2: Este almacenará el nombre del hospital 2 del departamento de Jutiapa conjuntamente a su numero de telefono.
- Hashmap jt3: Este almacenará el nombre del hospital 3 del departamento de Jutiapa conjuntamente a su numero de telefono.
- Hashmap jp1: Este almacenará el nombre del hospital 1 del departamento de Jalapa conjuntamente a su numero de telefono.
- Hashmap jp2: Este almacenará el nombre del hospital 2 del departamento de Jalapa conjuntamente a su numero de telefono.
- Hashmap jp3: Este almacenará el nombre del hospital 3 del departamento de Jalapa conjuntamente a su numero de telefono.
- Hashmap sr1: Este almacenará el nombre del hospital 1 del departamento de Santa Rosa conjuntamente a su numero de telefono.
- Hashmap sr2: Este almacenará el nombre del hospital 2 del departamento de Santa Rosa conjuntamente a su numero de telefono.
- Hashmap e1: Este almacenará el nombre del hospital 1 del departamento de Escuintla conjuntamente a su numero de telefono.

- Hashmap e2: Este almacenará el nombre del hospital 2 del departamento de Escuintla conjuntamente a su numero de telefono.
- Hashmap e3: Este almacenará el nombre del hospital 3 del departamento de Escuintla conjuntamente a su numero de telefono.
- Hashmap e4: Este almacenará el nombre del hospital 4 del departamento de Escuintla conjuntamente a su numero de telefono.
- Hashmap sc1: Este almacenará el nombre del hospital 1 del departamento de Sacatepéquez conjuntamente a su numero de telefono.
- Hashmap sc2: Este almacenará el nombre del hospital 2 del departamento de Sacatepéquez conjuntamente a su numero de telefono.
- Hashmap sc3: Este almacenará el nombre del hospital 3 del departamento de Sacatepéquez conjuntamente a su numero de telefono.
- Hashmap rl1: Este almacenará el nombre del hospital 1 del departamento de Retalhuleu conjuntamente a su numero de telefono.
- Hashmap rl2: Este almacenará el nombre del hospital 2 del departamento de Retalhuleu conjuntamente a su numero de telefono.
- Hashmap rl3: Este almacenará el nombre del hospital 3 del departamento de Retalhuleu conjuntamente a su numero de telefono.
- Hashmap sl1: Este almacenará el nombre del hospital 1 del departamento de Solola conjuntamente a su numero de telefono.
- Hashmap tt1: Este almacenará el nombre del hospital 1 del departamento de Totonicapán conjuntamente a su numero de telefono.
- Hashmap tt2: Este almacenará el nombre del hospital 2 del departamento de Totonicapán conjuntamente a su numero de telefono.
- Hashmap tt3: Este almacenará el nombre del hospital 3 del departamento de Totonicapán conjuntamente a su numero de telefono.
- Hashmap ch1: Este almacenará el nombre del hospital 1 del departamento de Chimaltenango conjuntamente a su numero de telefono.
- Hashmap ch2: Este almacenará el nombre del hospital 2 del departamento de Chimaltenango conjuntamente a su numero de telefono.
- Hashmap ch3: Este almacenará el nombre del hospital 3 del departamento de Chimaltenango conjuntamente a su numero de telefono.
- Hashmap such1: Este almacenará el nombre del hospital 1 del departamento de Suchitepéquez conjuntamente a su numero de telefono.

○ Métodos:

- Setguatemala: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Guatemala con su nombre y su numero de telefono al terminar este proceso agrega el HashMap de Guatemala con todos los hospitales de este departamento a la lista de hospitales.

- Setsanmarcos: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de San Marcos con su nombre y su numero de telefono al terminar este proceso agrega el HashMap de San Marcos con todos los hospitales de este departamento a la lista de hospitales.
- Sethuehuetenango: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Huehuetenango con su nombre y su numero de teléfono. Al terminar este proceso agrega el HashMap de Huehuetenango con todos los hospitales de este departamento a la lista de hospitales.
- Setquiche: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Quiché con su nombre y su número de teléfono al terminar este proceso agrega el HashMap de Quiche con todos los hospitales de este departamento a la lista de hospitales.
- Setaltaverapaz: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Alta Verapaz con su nombre y su numero de teléfono. Al terminar este proceso agrega el HashMap de Alta Verapaz con todos los hospitales de este departamento a la lista de hospitales.
- Setbajaverapaz: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Baja Verapaz con su nombre y su numero de telefono al terminar este proceso agrega el HashMap de Baja Verapaz con todos los hospitales de este departamento a la lista de hospitales.
- Setelprogreso: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de El Progreso con su nombre y su numero de telefono al terminar este proceso agrega el HashMap de El progreso con todos los hospitales de este departamento a la lista de hospitales.
- Setzacapa: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Zacapa con su nombre y su numero de telefono al terminar este proceso agrega el HashMap de Zacapa con todos los hospitales de este departamento a la lista de hospitales.
- Setizabal: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Izabal con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Izabal con todos los hospitales de este departamento a la lista de hospitales.
- Setpeten: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Petén con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de

Petén con todos los hospitales de este departamento a la lista de hospitales.

- Setchiquimula: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Chiquimula con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Chiquimula con todos los hospitales de este departamento a la lista de hospitales.
- Setjutiapa: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Jutiapa con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Jutiapa con todos los hospitales de este departamento a la lista de hospitales.
- SetJalapa: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Jalapa con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Jalapa con todos los hospitales de este departamento a la lista de hospitales.
- Setsantarosa: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Santa Rosa Con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Santa Rosa con todos los hospitales de este departamento a la lista de hospitales.
- Setescuintla: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Escuintla con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Escuintla con todos los hospitales de este departamento a la lista de hospitales.
- Setsacatepequez: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Sacatepequez con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Sacatepequez con todos los hospitales de este departamento a la lista de hospitales.
- Setretalhuleu: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Retalhuleu con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Retalhuleu con todos los hospitales de este departamento a la lista de hospitales.
- Setsolola: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Sololá con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Sololá con todos los hospitales de este departamento a la lista de hospitales.
- Settonicapan: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Totonicapán con su

nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Totonicapán con todos los hospitales de este departamento a la lista de hospitales.

- Setchimaltenango: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Chimaltenango con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Chimaltenango con todos los hospitales de este departamento a la lista de hospitales.
- Setsuchitepequez: Lo que realiza este método es configurar cada uno de los hospitales que conforman el departamento de Suchitepéquez con su nombre y su número de teléfono. Al terminar este proceso agrega el HashMap de Suchitepéquez con todos los hospitales de este departamento a la lista de hospitales.
- SetallHashmaps: Este metodo lo que realiza es definir todo lo que tiene un Hasmap dentro para poder hacer uso de los mismos.

- **Clase Interfazgeneral:**

- Descripción de la clase: Esta clase es una clase derivada de javax.swing.Jlabel y esta clase es la encargada de presentar de presentar la primera interfaz gráfica con sus diferentes botones los cuales serán los botones donde el usuario podrá ingresar a las diferentes opciones del programa y con esto a las diferentes interfaces del programa.
- Métodos:
 - jButton1ActionPerformed: Este método es el método para poder cerrar el programa completamente.
 - jButton2ActionPerformed: Este método es el método para poder acceder a la interfaz de Hospitales y sus diferentes métodos
 - jButton3ActionPerformed: Este método es el método para poder acceder a la interfaz de contactos de emergencia y su información
- Atributos:
 - Private labelfondo javax.swing.Jlabel: este atributo es para almacenar una imagen de fondo para la inetrfaz.
 - Private Jbutton1 javax.swing.Jlabel: Este atributo es para crear el botón que despliega la interfaz de hospitales.
 - Private Jbutton2 javax.swing.Jlabel: Este atributo es para crear el botón que despliega la interfaz de contactos de emergencia.
 - Private Jbutton3 javax.swing.Jlabel: Este atributo es para crear el botón que despliega será el botón de salida para acabar con la interfaz.
 - Private Jbutton3 javax.swing.Jlabel: Este atributo es para crear el botón que despliega la interfaz de sintomas.
 - Private Jbutton3 javax.swing.Jlabel: Este atributo es para crear el botón que despliega la interfaz de probabilidad de riesgo.

- Private JButton3 javax.swing.JLabel: Este atributo es para crear el botón que despliega la interfaz Display1.
- Private JLabel1 javax.swing.JLabel: Este atributo es para crear el label en donde estará guardado el icono del programa.

- **Clase Principal:**

- Descripción de la clase: Esta clase es una clase derivada de javax.swing.JLabel y esta clase es la encargada de presentar la interfaz gráfica del directorio de hospitales. También es la encargada de interactuar con el usuario y presentarle todo de una manera más amigable.

- Atributos:

- private javax.swing.JTextField CampoN1: Este atributo es el encargado de recibir el input del usuario dentro de la interfaz gráfica.
- private javax.swing.JLabel JLabelFondo: Este atributo es un label que contiene el fondo de la interfaz que se crea para que esta sea amigable con los usuarios.
- private javax.swing.JButton jButton1: Esta variable se crea para identificar al boton que sera el boton para salir de la interfaz.
- private javax.swing.JButton jButtonConfirmar: Este atributo se crea para identificar el boton para salir de esta interfaz a la interfaz general.
- private javax.swing.JLabel jLabel1: Esta variable se crea para poder almacenar uno de los textos presentados al usuario.
- private javax.swing.JLabel jLabel2: Esta variable se crea para poder almacenar uno de los textos presentados al usuario.
- private javax.swing.JLabel jLabel3: Esta variable para identificar al label que contiene el mapa que es para que el usuario se identifique.
- private javax.swing.JScrollPane jScrollPane1: Esta variable se crea para identificar el panel donde se genera la información extraída por el usuario cuando selecciona algún departamento.

- Métodos:

- Public principal: Este método es el método el cual inicializa la interfaz gráfica e inicializa la clase directorio.
- private void jButton1ActionPerformed: Este es el método que hace que termine la interfaz y se redirige a la interfaz más general.
- private void CampoN1ActionPerformed: Este es el método que recibe la información del usuario para luego ser utilizada en los métodos de interacción.
- private void jButtonConfirmarActionPerformed: Este es el método que ejecuta las acciones y luego devuelve el String a la caja de texto para que esta sea presentada.

- **Clase Principal1:**

- Descripción de la clase: Esta clase es una clase procedente de javax.swing.JLabel, además es la encargada de mostrar la interfaz gráfica de la

clase que le solicita al usuario sus síntomas; también se encarga de interactuar con el usuario, para que este aprecie todo más amigable.

- **Atributos:**

- private javax.swing.JTextField CampoN1: Este atributo es el encargado de recibir el String (síntomas) del usuario dentro de la interfaz gráfica.
- private javax.swing.JButton jbtnSalir: variable que se crea para identificar al botón que permitirá salir de la interfaz.
- private javax.swing.JButton jbtncontinuar: Este atributo se crea para identificar el botón continuar, quién es el que almacena el dato que el usuario ingresa.
- private javax.swing.JLabel j: Esta variable se crea para poder almacenar uno de los textos presentados al usuario.
- private javax.swing.JLabel jLabel2: Esta variable se crea para poder almacenar uno de los textos presentados al usuario.
- private javax.swing.JScrollPane jcampoN2: Esta variable se crea para identificar el panel donde se muestran los tips que se pueden realizar en casa con productos naturales, además se muestran los medicamentos recomendados incluyendo dosis, dependiendo en la categoría de síntomas en la que se encuentre (usuales, inusuales y graves).

- **Métodos:**

- public principal: Este método se encarga de inicializar la interfaz gráfica.
- private void jButton1ActionPerformed: Este es el método que hace que termine la interfaz y se redirige a la interfaz más general.
- private void CampoN1ActionPerformed: Este es el método que recibe la información del usuario para luego ser utilizada en los métodos de interacción.
- private void jButtonConfirmarActionPerformed: Este es el método que ejecuta las acciones y luego devuelve el String a la caja de texto para que esta sea presentada.

- **Clase OperacionesS:**

- Descripción: clase que contiene el método encargado de solicitar al usuario los síntomas que presenta, además se encarga de retornar un String donde muestra los tips y medicamentos que puede realizar según en la categoría en la que se encuentra.

- **Atributos:**

- sintomasComunes: array que almacena los síntomas más comunes que puede presentar el usuario.
- sintomasMenosComunes: array que almacena los síntomas menos comunes que puede presentar el usuario.
- sintomasGraves: array que almacena los síntomas graves que puede presentar el usuario.

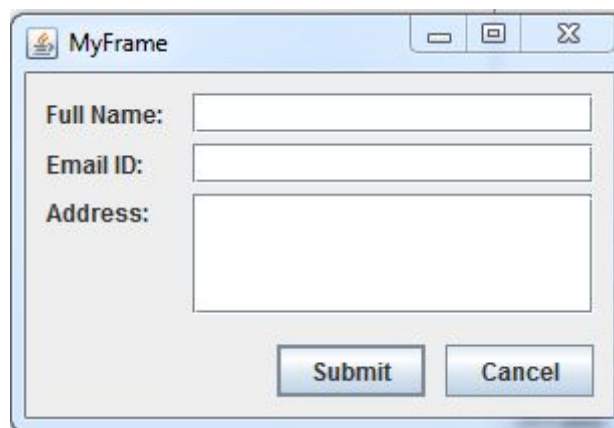
- **Métodos:**

- IngresarSintomas: método encargado de solicitarle al usuario sus síntomas y ubicarlos en una categoría y luego retornar una serie de consejos que puede realizar, tanto naturales como posibles medicamentos que éste puede tomar.

Uso de herencia dentro del programa

Para la realización de la interfaz gráfica en nuestro programa se utilizó herencia para extender la clase JFrame de la biblioteca Swing para poder generar las diferentes ventanas y elementos que serían parte de nuestra interfaz. Esta clase permite que modifiquemos la ventana con elementos como botones, textos y nos permite hacer funciones básicas de una ventana como minimizar, cerrar, maximizar y moverla por la pantalla.

Ejemplo de una ventana hecha con JFrame:



Uso de archivos Jar para el programa

Pasamos el programa a un formato de archivo Jar, para que se pueda correr sin ningún tipo de inconveniente o error por la línea de comandos (CMD)

Se decidió pasarlo a este formato de archivo ya que varias clases hacían uso de varios métodos y componentes de la librería externa `AbsoluteLayout` que facilita la creación de interfaces gráficas como la creada para esta entrega.

La librería `AbsoluteLayout` fue necesaria para algunas de las clases ya que nos permite poner objetos de las interfaces más libremente dentro de la ventana que se le presenta al usuario, ya que otros tipos de layouts de interfaces como `CardLayout`, `BoxLayout` y `GridLayout` son

muy limitantes en cuanto a cómo se le permite al desarrollador poner los diferentes elementos dentro de la interfaz.

Aplicacion de la tecnologia disponible

Planificacion y gestion

1. Ejemplo

- a. Descripción:
- b. Horas de desarrollo:
- c. Responsables del desarrollo:
- d. Fecha probable de terminación:

1. Interfaz Gráfica Contactos Emergencia

- a. Descripción: Implementar la interfaz gráfica a la clase de ContactosEmergencia para lograr mostrar una interfaz más amigable para el usuario.
- b. Horas de desarrollo: 1 hora
- c. Responsables del desarrollo: Andres de la Roca
- d. Fecha probable de finalizacion: 3/11/2020

2. Interfaz gráfica general e Interfaz gráfica Directorio, Actualización de la clase Directorio para que esta se desarrolle de una mejor manera.

- a. Implementar la Interfaz general para poder generar un mejor orden entre las interfaces y así hacer nuestro programa más amigable para el usuario. Implementar la interfaz gráfica para el directorio de hospitales para que esta pueda interactuar con el usuario de una mejor manera y sé amigable al momento de ingresar los datos, también se realizó una modificación de la misma para que esta no presenta ningún warning y los datos estuvieran organizados de una mejor manera.
- b. Horas de desarrollo:4
- c. Responsables: Pablo Gonzalez
- d. Fecha de finalización: 3/11/2020

3. Trabajo Escrito (Informe):

- a. Descripción: Escribir el informe de la entrega 4 del proyecto, cumpliendo todos los requerimientos de la guía.
- b. Horas de desarrollo: Aproximadamente de 3 a 6 horas.
- c. Responsables del desarrollo: Todos los integrantes del grupo.

- d. Fecha probable de Finalización: 2/11/2020.
4. Interfaz gráfica Síntomas, Actualización de la clase Sintomas para que esta se ejecute de forma más eficaz.
- Desarrollar la interfaz gráfica de los síntomas, para que la interacción del usuario y el programa sea más amigable.
 - Desarrollo del UML actualizado con los cambios respectivos en cada clase, como agregar las nuevas clases que permiten que se ejecute la interfaz.
 - Horas de desarrollo: 3 horas
 - Responsables: Estefanía Elvira
 - Fecha de finalización: 4/11/2020.

Nombre: Andrés de la Roca

Carné: 20332

Fecha	Inicio	Fin	Tiempo interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
4/11/2020	1/10/2020	4/11/2020	15	60	Interfaz Gráfica de clase Contacto Emergencia	
4/11/2020	1/10/2020	4/11/2020	30	120	Trabajo Escrito (Informe)	

Nombre: Estefania Ramos

Carné: 20725

Fecha	Inicio	Fin	Tiempo interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
04/11/2020	1/10/2020	4/11/2020	15	90	Interfaz gráfica clase Sintomas	
04/11/2020	1/10/2020	4/11/2020	25	120	Trabajo Escrito	
5/10/2020	5/11/2020	5/11/2020	20	80	Desarrollo del UML	

Nombre: Pablo Gonzalez

Carné: 20362

Fecha	Inicio	Fin	Tiempo interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
3/11/2020	29/09/2020	3/11/2020	30	240	Creación de interfaz gráfica para el área de hospitales y actualización de la clase Directorio	

Nombre: Cristian Aguirre

Carné: 20231

Fecha	Inicio	Fin	Tiempo interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
4/11/2020	4/11/2020	4/11/2020	30	190	Creación de interfaz gráfica para la calculadora de riesgo	
5/11/2020	5/11/2020	5/11/2020	60	150		

Nombre: Manuel Archila

Carné: 161250

Fecha	Inicio	Fin	Tiempo interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
4/11/2020	5:00 p.m.	10p.m.	90min	210min	Crear la interfaz gráfica para las maneras, síntomas, cuidados y	

					referencias externas	
--	--	--	--	--	-------------------------	--

Nombre: Jun Woo Lee Carné: 20358

Fecha	Inicio	Fin	Tiempo interrupción (min)	Delta Tiempo (min)	Tarea	Comentarios
04/11/2020	2:00	6:00	60 minutos	3 horas	Crear la interfaz gráfica para el sistema de log in y registro	
05/11/2020	5:00	6:00	0 minutos	1 hora	Trabajo escrito	

Matriz de Retroalimentación:



Figura 1. Organización de la matriz de retroalimentación

Nombre: Melanie Morales
Edad: 20

Carrera: Ingeniería Mecatrónica

Método de contacto: Zoom

El usuario comenta que le gustó programa, pero hizo énfasis en aclarar su agrado con la veracidad de la información, especialmente en la sección de contactos de emergencia	Crítica los colores de la interfaz, le parecen muy apagados, además hacen que la interfaz no se vea del todo estética.
El usuario no tuvo ninguna duda	El usuario sugiere que se agregue una opción para poder conocer a los demás usuarios registrados, además indicó que le gustaría poder comunicarse con el resto de usuarios.

Nombre: Hugo Elvira

Edad: 23

Carrera: Ingeniería Electrónica

Método de contacto: Zoom

El usuario comenta que le pareció que la información que muestra el programa es muy importante y le gustó mucho que muestre el porcentaje de riesgo según la edad e historial médico.	El usuario criticó que el nombre, colores y logo de la interfaz no le parecieron los recomendados, además indicó que no son tan llamativos
El usuario no tuvo ninguna duda.	El usuario sugiere que se haga una opción para darle seguimiento durante todo el tiempo que tengan sus síntomas, además indica que sería ideal que en el programa se pueda hacer como un horario de las medicinas que debe tomar y que este le avise cuando sea hora de tomarlas.

Nombre: Sol de León

Edad: 18

Carrera: Diseño Gráfico

Método de contacto: Zoom

El usuario comenta que la información presentada por el programa es información muy útil y siempre se utilizan los términos adecuados para poder presentar la información. El usuario comenta que le gusto que en el inicio de sesión este prevea errores ya que se siente seguro con sus datos de ingreso.	El usuario critica la fuente tipográfica debido a que las letras están muy juntas y es complicado de leer, también comenta que cambiaría la fuente
Al usuario no le generó ninguna pregunta el programa ya que cree que todo está muy claro al momento de usar el mismo.	El usuario cambiaría los colores de la interfaz ya que piensa que estos son colores muy saturados.

Nombre: Nathaly Gonzalez

Edad: 17 años

Carrera: Arquitectura

Método de contacto: Zoom

El usuario comenta que le gusto que el programa está organizado y que la interfaz gráfica le pareció excelente y que la información es muy fácil de entender. El usuario comenta que el programa presenta información útil para saber que es más a fondo del Covid 19.	El usuario comenta que para su gusto el logo es algo que no le parece muy llamativo.
El usuario comenta que no le generó ninguna pregunta.	El usuario le gustaría una pestaña extra en donde se puedan ver los contagios por departamento.

Nombre: Miguel Castillo

Edad: 19

Carrera: Ingeniería Civil

Método de contacto: Llamada en Discord

El usuario cuenta que la información se muestra de una manera muy amigable debido a la interfaz gráfica, adicionalmente comenta que le gusto mucho la información recopilada que esta puede ayudar a generar conciencia en las personas de que se tienen que cuidar del COVID-19.	El usuario critica que el esquema de colores usado para la interfaz es muy oscuro y apagado.
Pregunta si hay algunas otras funcionalidades que están planeadas para	Comenta que se le podrían añadir elementos mas relacionados con el COVID-19 (Como

agregarle al programa.	imágenes y elementos de la interfaz) para que el usuario entre en mejor contexto de lo que se trata el programa.
------------------------	--

Nombre: Melany Perez

Edad: 19

Carrera: Derecho

Método de contacto: Whatsapp

Le gusta la funcionalidad de determinar cuanta probabilidad tiene alguien de contagiarse del COVID-19 según la edad y enfermedades que tenga alguien.	El usuario crítica que la interfaz no debería de ser en pantalla completa y más bien una ventana que se pueda minimizar y maximizar como lo desee el usuario.
Pregunta si se agregaran mas tipos de síntomas a la sección de detección de síntomas.	Comenta que se podría agregar una funcionalidad para enviar sugerencias a los desarrolladores para así mejorar el programa.

Nombre: Ignacio Sanchez Parra

Edad: 19

Carrera: Letras

Método de contacto: Whatsapp

Le gusta la facilidad de encontrar los hospitales que aportan ayuda contra covid.	El usuario crítica que no hay mucha flexibilidad para ingresar los síntomas
Pregunta si el diseño del log in va a quedar igual.	Comenta que no está estéticamente agradable

Nombre: David Narciso

Edad: 19

Método de contacto: Llamada de Zoom

Le gusta la calculadora de riesgo que muestra diferentes simulaciones de opciones	El usuario crítica que la interfaz es muy grande, cuando no necesita de ser tan grande.
Pregunta si van a agregar más síntomas al programa.	Comenta que la funcionalidad le parece muy útil. Y que si fuera una aplicación para móvil lo utilizaría.

Nombre: Nicolas Maldonado
Edad: 21
Carrera: Arte
Método de contacto: Whatsapp

Le gusto que tiene mucha información y también muestra links de sitios confiables ya que en estos tiempos hay mucha información falsa.	El usuario crítica que el sistema de síntomas es muy restringido y difícil de usar.
El usuario pregunta si piensan expandir sobre más allá de solamente Guatemala.	.El usuario comenta que le gusta el diseño de la interfaz gráfica.

Nombre: Juan Pablo Mapape
Edad: 25
Carrera: Finanzas
Método de contacto: Videollamada de Skype

Le gusto que la interfaz fuera amigable con el usuario y que la distribución de las opciones fuera uniforme.	El nombre del programa le parece muy cliché, propone que se le cambie el nombre
El usuario pregunta si en algún momento se le agregan opciones interactivas con el usuario, como lo podrían ser preguntas trivias, etc.	Pregunta que si se le agregara alguna opción de videos informativos.

Nombre: Josselyn Aguirre
Edad: 27 años
Carrera: Medica Cirujana
Método de contacto: Zoom

Le gusta el diseño de la interfaz gráfica pues considera que es bastante llamativa y amigable.	El usuario criticó que en algunas opciones el tamaño de la fuente no es el adecuado y puede afectar cuando alguna persona con problemas de visión use el programa.
El usuario considera que todo el programa es claro por lo que no tiene ninguna pregunta	Comenta que en el cálculo de probabilidad se podría agregar la condición física del paciente pues es un aspecto importante a evaluar para el cálculo de riesgo.

Nombre: Amilcar Palma

Edad: 18 años

Carrera: Ingeniería en Sistemas

Método de contacto: Zoom

Le gusta la facilidad de uso que presenta el programa pues está todo muy bien indicado y no cuesta manejarlo.	Considera que las opciones de síntomas no son suficientes para poder identificar de mejor manera la enfermedad.
Pregunta si ese será el diseño final de la interfaz gráfica o se le harán cambios.	Comenta que es un programa bastante completo y que puede ser de gran ayuda en la actualidad que estamos viviendo, únicamente piensa que se deben afinar detalles menores de índole estético.

Control de versiones

<https://github.com/andresdlRoca/Proyecto2doSemestre>

Referencias

- How To: The Comma Separated Value File Format (2010) Dominic Repici.
<http://www.creativyst.com/Doc/Articles/CSV/CSV01.htm>
- Interfaz Gráfica de Usuario Con Netbeans (2017) Gaby Nieva.
<https://dcodingames.com/interfaz-grafica-de-usuario-con-netbeans/>
- Swing - Java (2020) Diego Molaset.
<https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?codigo=104&punto=&inicio=20>