Universidad Nacional de Colombia - sede Medellín
Facultad de Minas
Ingeniería de Sistemas
Curso: POO

Estudiante: Sebastián Soto Arcila
Correo: ssotoa@unal.edu.co
CC: 1000438004

Mensajería

Descripciones generales de la solución

El programa parte de un usuario el cual tiene su propio **Contacto de Usuario**, donde están todos sus datos y los de su negocio. Además, tiene dos tipos de listas de contactos:

- **Contactos pendientes:** personas con las que aún no se ha iniciado una conversación.
- **Contactos locales:** personas con las que ya se inició una conversación.

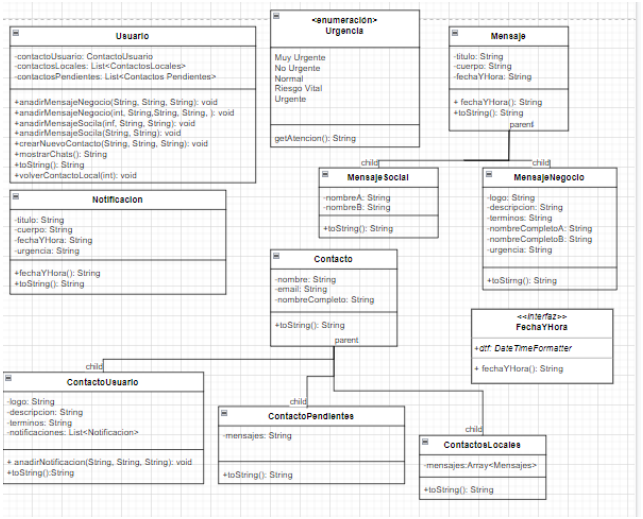
Las conversaciones con los contactos se realizan por medio de mensajes que pueden ser de dos tipos:

- **Mensajes Sociales:** son mensajes informales, solo tienen fecha, hora, título y cuerpo.
- **Mensajes de Negocios:** son mensajes formales, tienen la información de un mensaje social y también la información sobre el negocio del usuario. También implementan un dato que informa su nivel de urgencia.

El usuario puede enviarse mensajes a sí mismo, esto lo hace por medio de las **Notificaciones**. Estas están conformadas por una fecha, hora, título, cuerpo y nivel de urgencia.

Los niveles de **Urgencia** de las notificaciones y mensajes de negocios son los siguientes, riesgo vital, muy urgente, urgente, normal, y no urgente. con un tiempo estimado de respuesta de Inmediatamente, 10 a 15 minutos, 60 minutos, 2 horas y 4 horas respectivamente.

Descripción del diseño UML



Descripción de la Implementación

- Las clases Contacto y Mensaje son abstractas, tienen los atributos que todas las clases hijas deben tener. La clase Mensaje tiene el método abstracto toString para obligar a las clases hijas a sobreescribirlo.
- Se implementa la interfaz FechaYHora que tiene un atributo que indica el formato en el que se va a mostrar la fecha y hora, y un metodo fechaYHora. lo implementan MensajeNegocios, MensajeSocial y Notificacion, para guardar la fecha y hora de creación de una instancia de estos.
- Las Clases ContactosLocales, ContactosPendientes y ContactoUsuario, heredan de Contacto. Lo mismo sucede con MensajeSocial y MensajeNegocios, que heredan de Mensaje.
- El método de instancia mostrarChats de la clase Usuario, guarda el retorno del toString de los mensajes de cada contacto local, el apuntador es de tipo Mensaje, pero al llamar el método toString lo ejecuta de las clase más específicas MensajeSocial o MensajeNegocios.
- Las Clases ConexionUsuario, ConexionContactoUsuario y DatosContactoUsuario, tienen los atributos usuario, contactoUsuario y datos, respectivamente, ya que de esta forma se evita ambigüedad a la hora de llamar a estas instancias. La clase conexionUsuario utiliza el método estático conexionUrgencia para que esta y otras clase puedan acceder a la enumeración Urgencias. Y la Clase ConexionContactoUsuario tiene los métodos cargarContactoUsuario y guardarContactoUsuario, que se encargan de la persistencia; la clase DatosContactoUsuario tiene dos métodos estáticos que son similares a los métodos anteriores.
- La instancias de las clase Notificacion y MensajeNegocios, tiene un atributo fechaYHora que se encarga de guardar la hora y fecha de creación de estas instancias. Es importante que este dato no se modifique, por eso se trata como constante.
- Todas las clases del paquete gestorAplicacion.inferior son de tres tipos de acceso. Las públicas sólo son Usuario y ContactoUsuaro ya que se comunican con los paquetes uiMain y baseDatos, las clases protected son las que heredan de del paquete gestorAplicaciones.superior, y las demás clase son de paquete. Los métodos de estas clases siguen este mismo sistema y los atributos en general son privados.
- La clase MensajeSocial tiene sobrecarga de constructor, y las clase Usuario tiene sobrecarga en los métodos anadirMesajesSocial y anadirMensajeNegocios.
- Se usa el this en todas las clases en constructos y el los métodos getters y setter que no son heredados. Y el this() en la clase MensajeSocial.
- Se utiliza la enumeración Urgencias.
- La persistencia solo se implemento para la instancia ContactoUsuario.

Descripción de las funcionalidades

```
Programa de Mensajeria
Opciones:
        0 Salir del Programa
        1 Editar Perfil
        2 Agregar un Nuevo Contacto
        3 Empezar una Conversacion
        4 Mostrar Chats
        5 Seguir una Conversacion
        6 Ver Notificaciones de Usuario

Opcion: 1
```

1. **Editar Perfil:** permite al usuario editar su contacto de usuario y la información de su negocio. En esta funcionalidad participan las clases Pantalla, ConexionContactoUsuario y ContactoUsuario, de la siguiente forma:
    a. **Pantalla** mediante la instancia de **ConexionContactoUsuario** llama al método editarPerfil.
    b. **ConexionContactoUsuario** a través de la instancia de **ContactoUsuario** y usando los getter y setter hace las modificaciones a la instancia.

```
-----Editar Perfil-----
Contacto [
        Nombre: Andres
        Email: Andres@a.com
        Nombre Completo: Andres Acosta Aguirre
]
Tipo: ContactoUsuario [
        Logo del Negocio: Andidas
        Descripcion del Negocio: No vendo zapatos...
        Terminos del Negocio: solo efectivo
]
Opciones:
        0 Terminar Edicion
        1 Editar Nombre
        2 Editar Email
        3 Editar Nombre Completo
        4 Editar Logo del Negocio
        5 Editar Descripcion del Negocio
        6 Editar Terminos del Negocio

Opcion: 6

Terminos de negocio actual: solo efectivo
Nuevo terminos de negocio: Solo efectivo y Transferencia BancoLoco
-----Editar Perfil-----
```

2. **Agregar un Nuevo Contacto:** crea y agrega un nuevo contacto pendiente. Usa las clase Pantalla, ConexionUsuario, Usuario y ContactosPendientes, así:
    a. **Pantalla** utiliza un método de instancia de la clase **ConexionUsuario** llamado crearNuevoContacto, en donde se crea un objeto tipo **ContactosPendientes** y se agrega al atributo contactosPendietes de la instancia de **Usuario**.
    b. Este método antes mencionado muestra los contactos que están en contactosPendientes antes de que el usuario cree un nuevo contacto.

```
-----Crear Contacto-----
[Contacto [
        Nombre: Alfredo
        Email: Alfredo@unal.edu.co
        Nombre Completo: Alfredo Ramos
]       Tipo: ContactosPendientes [
        mensajes :No hay mensajes
]
, Contacto [
        Nombre: Daniel
        Email: Daniel@gmail.com
        Nombre Completo: Daniel Soto
]       Tipo: ContactosPendientes [
        mensajes :No hay mensajes
]
]
Nuevo Contacto
        Nombre: Julio
        Email: JJaramillo@Yahoo.com
        Nombre completo: Julio Jaramillo
-----Contacto Creado-----
```

3. **Empezar una Conversación:** permite enviar mensajes a los contactos que aún no tienen mensajes. Las clases necesarias son Pantalla, ConexionUsuario, Usuario, ContactosLocales, ContactosPendientes, MensajeLocales y MensajeSociales, tambien la enumeración Urgencia y la interfaz FechaYHora. de la siguiente manera:
    a. **Pantalla** llama al método empezarChat de la instancia de **ConexionUsuario**, éste cambia el contacto elegido de **ContactosPendientes** a **ContactosLocales**, con ayuda del método volverContactoLocal de instancia de **Usuario** y lo añade al atributo contactosLocales.
    b. Luego empezarChat llama al método crearMensaje de la clase **ConexionUsuario**, el cual según la decisión de usuario, crea un **MensajeSocial** o **MensajeNegocios,** con los métodos anadirMensajeSocial y anadirMensajeNegocios de las instancia usuario. Este mensaje se añade al atributo mensajes del nuevo **ContactosLocales**.
    c. Si el método escogido en el paso anterior es **MensajeNegocios**, antes de crear el objeto, con ayuda del método conexionUrgencia que utiliza la enumeración **Urgencia** se elige su urgencia.

```
-----Empezar Chat-----
Elige un contacto o Cancela:
        0 Cancelar

        1 Contacto [
        Nombre: Alfredo
        Email: Alfredo@unal.edu.co
        Nombre Completo: Alfredo Ramos
]       Tipo: ContactosPendientes [
        mensajes :No hay mensajes
]

        2 Contacto [
        Nombre: Daniel
        Email: Daniel@gmail.com
        Nombre Completo: Daniel Soto
]       Tipo: ContactosPendientes [
        mensajes :No hay mensajes
]

        3 Contacto [
        Nombre: Julio
        Email: JJaramillo@Yahoo.com
        Nombre Completo: Julio Jaramillo
]       Tipo: ContactosPendientes [
        mensajes :No hay mensajes
]

Opcion: 1
Titulo: Hola
Cuerpo: ¿Como estas?
Elige el tipo de mensaje:
        0 Negocios
        1 Social

Opcion: 1
-----Empezar Chat-----
```

```
-----Empezar Chat-----
Elige un contacto o Cancela:
        0 Cancelar

        1 Contacto [
        Nombre: Daniel
        Email: Daniel@gmail.com
        Nombre Completo: Daniel Soto
]       Tipo: ContactosPendientes [
        mensajes :No hay mensajes
]

        2 Contacto [
        Nombre: Julio
        Email: JJaramillo@Yahoo.com
        Nombre Completo: Julio Jaramillo
]       Tipo: ContactosPendientes [
        mensajes :No hay mensajes
]

Opcion: 2
Titulo: Contrato Musical
Cuerpo: Quiero que cante en el aniversario de la compañia
Elige el tipo de mensaje:
        0 Negocios
        1 Social

Opcion: 0
Tipo de Urgencia:
        Opciones:
                1 Riesgo Vital
                2 Muy Urgente
                2 Urgente
                4 Normal
                5 No Urgente

Tipo: 2
-----Empezar Chat-----
```

4. **Mostrar Chats:** muestra los mensajes creados por cada contacto, para esta tarea se usan las clases Pantalla, ConexionUsuario, Usuario, ContactosLocales y Mensajes:
    a. **Pantalla** llama al método mostrarChat de la instancia de **ConexionUsuario**, que también llama a un método con el mismo nombre de la instancia **Usuario**.
    b. Este último método recorre los **ContactosLocales** en el atributo con el mismo nombre del **Usuario**, y guarda los mensajes de cada contacto en un String.
    c. El String se imprime desde la clase **ConexionUsuario** en el método mostrarChat.

```
-----Chats-----
Email: Alfredo@unal.edu.co
        Mensajes:
        MensajeSocial [
        Fecha Y Hora: 2022/10/25 10:25:03
        Titulo: Hola
        Cuerpo: ☐Como estas?
        Nombre Usuario: Andres
        Nombre Contacto: Alfredo
]
Email: JJaramillo@Yahoo.com
        Mensajes:
        MensajeNegocios [
        Fecha Y Hora: 2022/10/25 10:29:43
        Titulo: Contrato Musical
        Cuerpo: Quiero que cante en el aniversario de la compa☐ia
        Urgencia:10 - 15 Minutos
        Logo del Negocio: Andidas
        Descripcion del Negocio:No vendo zapatos...
        Terminos del Negocio: Solo efectivo y Transferencia BancoLoco
        Nombre Completo Usuario: Andres Acosta Aguirre
        Nombre Completo Contacto: Julio Jaramillo
]

-----Estos fueron todos los Chats-----
```

5. **Seguir una Conversación:** permite crear mensajes a contactos con los que ya se ha iniciado una conversación, usa las clases Pantalla, ConexionUsuario, Usuario, ContactosPendientes, MensajeLocales y MensajeSociales, también la enumeración Urgencia

y la interfaz FechaYHora. Funciona igual a la funcionalidad 3, solo que en esta se salta el paso de convertir un contacto.

```
-----Seguir Chat-----
Elige un contacto o Cancela:
        0 Cancelar

        1 Contacto [
        Nombre: Alfredo
        Email: Alfredo@unal.edu.co
        Nombre Completo: Alfredo Ramos
]
        2 Contacto [
        Nombre: Julio
        Email: JJaramillo@Yahoo.com
        Nombre Completo: Julio Jaramillo
]
Opcion: 2
Titulo: Muchas Gracias
Cuerpo: El concierto buenisimo
Elige el tipo de mensaje:
0 negocios
1 Social
1
-----Seguir Chat-----
```

6. **Ver Notificaciones de Usuario:** tiene las funcionalidad de ver las notificaciones existentes y crear nuevas, usando las clases Pantalla, ConexionContactoUsuario, ContactoUsuario y Notificacion, de esta manera:
    a. **Pantalla** llama al método verNotificaciones de instancia **ConexionContactoUsusario,** esta muestra las notificaciones ya creadas, y le pide al usuario la información para crear la nueva notificación.
    b. verNotificaciones, usa el metodo de instancia **ContactoUsuario** anadiraNotificaciones para crear la **Notificación** y añadirla al atributo notificaciones de **ContactoUsuario**.

```
-----Crear Notificaciones-----
Notificaciones:
        Notificacion [
        Fecha y Hora: 2022/10/25 10:48:43
                Titulo: Lavar carro
                Cuerpo: Esta muy Sucio
                Urgencia: 60 Minutos
]

Crear una nueva:
        0 No
        1 Si

Opcion: 1
titulo: Hablar con Julio
cuerpo: Perdirle que cante en el aniversario
Tipo de Urgencia:
        Opciones:
                1 Riesgo Vital
                2 Muy Urgente
                3 Urgente
                4 Normal
                5 No Urgente

Tipo: 1
```

Manual de Usuario

Al iniciar la aplicación, las funciones disponibles son: editar perfil (1), agregar nuevo contacto (2) y ver notificaciones (6). se puede acceder a ellas ingresando su número.

Después de agregar un nuevo contacto, se puede usar la funcionalidad empezar una conversación (3).

Después de empezar una conversación, se puede seguir una conversación (5).

Teniendo en cuenta lo anterior. Y que el programa va a fallar cuando espera recibir números o cuando espera recibir una palabra(sin espacios) como por ejemplo Email, y se le pone un valor diferente a estos.

Estas son todas las opciones que tiene el programa (para utilizarlas debe poner el número correspondiente en el menú principal.):

- **Editar perfil (1):** muestra el estado del contacto de usuario y permite elegir qué atributo se desea modificar con un selector del 1 al 6,y luego ingresar el nuevo valor. Para terminar la edición es necesario ingresar 0 (Los cambios quedarán guardados al cerrar el programa).
- **Agregar nuevo contacto (2):** al entrar en esta funcionalidad se pedirá ingresar nombre, luego email y por último nombre completo.
- **Ver notificaciones (6):** permite ver las notificaciones creadas, y elegir si se desea crear una (ingresar 1 para crearla y 0 para salir de la funcionalidad). Si se elige crear una nueva, primero debe ingresar título de la notificación, luego su cuerpo y por último elegir entre 1 y 5 en nivel de urgencia.
- **Mostrar chats (4):** muestra los chat separados por dirección de Email.
- **Empezar una Conversación (3) y Seguir una Conversación (5):** en ambas debe primero seleccionar un contacto mediante su número, ingresar el título de mensaje, luego su cuerpo y por último elegir el tipo de mensaje (0 para negocios y 1 para social). **Si el mensaje es de negocios** también se pedirá elegir entre 1 y 5 el nivel de urgencia. **Para terminar este proceso ingrese 0 en la selección de contactos**.