

Taller (Directorio de Amistades)

María Paz Puerta Acevedo

Heily Yohana Rios Ayala

Carlos Andres Rojas Paez

Programación Orientada Objetos

Universidad Autonoma de Manizales

26/03/2025

Grupos de trabajo

Este taller se desarrollará en grupos de dos estudiantes.

Enunciado

Los estudiantes de UAM, como son personas tan amables y queridas por la comunidad, tienen muchos amigos y eso les dificulta llevar un completo registro de ellos y de sus informaciones de contacto. Por eso, se propone crear una aplicación para registrar la información de contacto de todos nuestros amigos.

De un amigo se conoce la siguiente información: nombres, teléfono y dirección de correo electrónico. La aplicación deberá permitirle al usuario realizar las siguientes tareas:

- Agregar un nuevo amigo.
- Buscar a un amigo previamente registrado (por correo electrónico).
- Ver el "Acerca de" esta aplicación (un diálogo con el nombre del taller, la asignatura, la universidad y los nombres de los desarrolladores).
- Salir de la aplicación.

Para **agregar un nuevo amigo** se debe tener en cuenta lo siguiente:

- Todos los datos son obligatorios; si falta alguno, no podrá registrarse.
- La dirección de correo debe tener una arroba (@) (se puede usar `indexOf` de `String`).
- Los teléfonos deben iniciar por los dígitos 606 o 30; de lo contrario, serán inválidos (se puede usar `startsWith` de `String`).
- No se pueden duplicar registros. La dirección de correo electrónico será el identificador único.

Para **buscar un amigo** se debe tener cuenta que:

- Se utilizará la dirección de correo electrónico para buscarlo.
- Si el amigo no está registrado, se mostrará un mensaje de advertencia al usuario.

Observaciones

Para el desarrollo del taller, se deben seguir buenas prácticas de programación. En especial, evitar mezclar diferentes lógicas.

- Los intentos fallidos al registrar o buscar amigos deben manejarse con excepciones propias.
- Analizar cuidadosamente quién detecta el problema y lanza la excepción, y quién debe manejarla.

Entregables

1. Diagrama de clases.
2. Listado de excepciones propias implementadas.
3. Implementación de la aplicación y presentación final.

1. Identificación de clases

- Amigo

Atributos

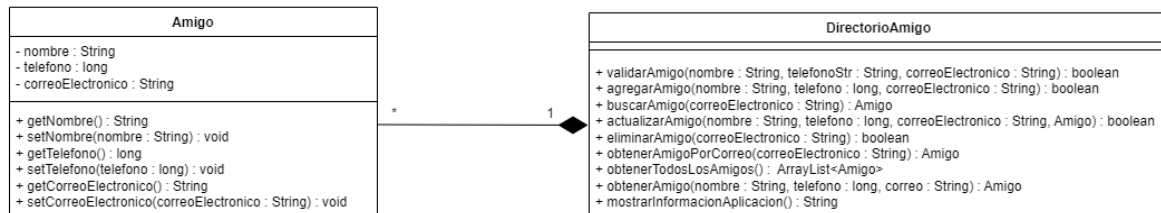
- ☐ nombre : String
- ☐ telefono : long
- ☐ correoElectronico : String

- DirectorioAmigo

Métodos

- ☐ agregarAmigo(nombre : String, telefono : long, correoElectronico : String, Amigo) : boolean
- ☐ buscarAmigo(correoElectronico : String) : Amigo
- ☐ mostrarInformacionAplicacion() : String

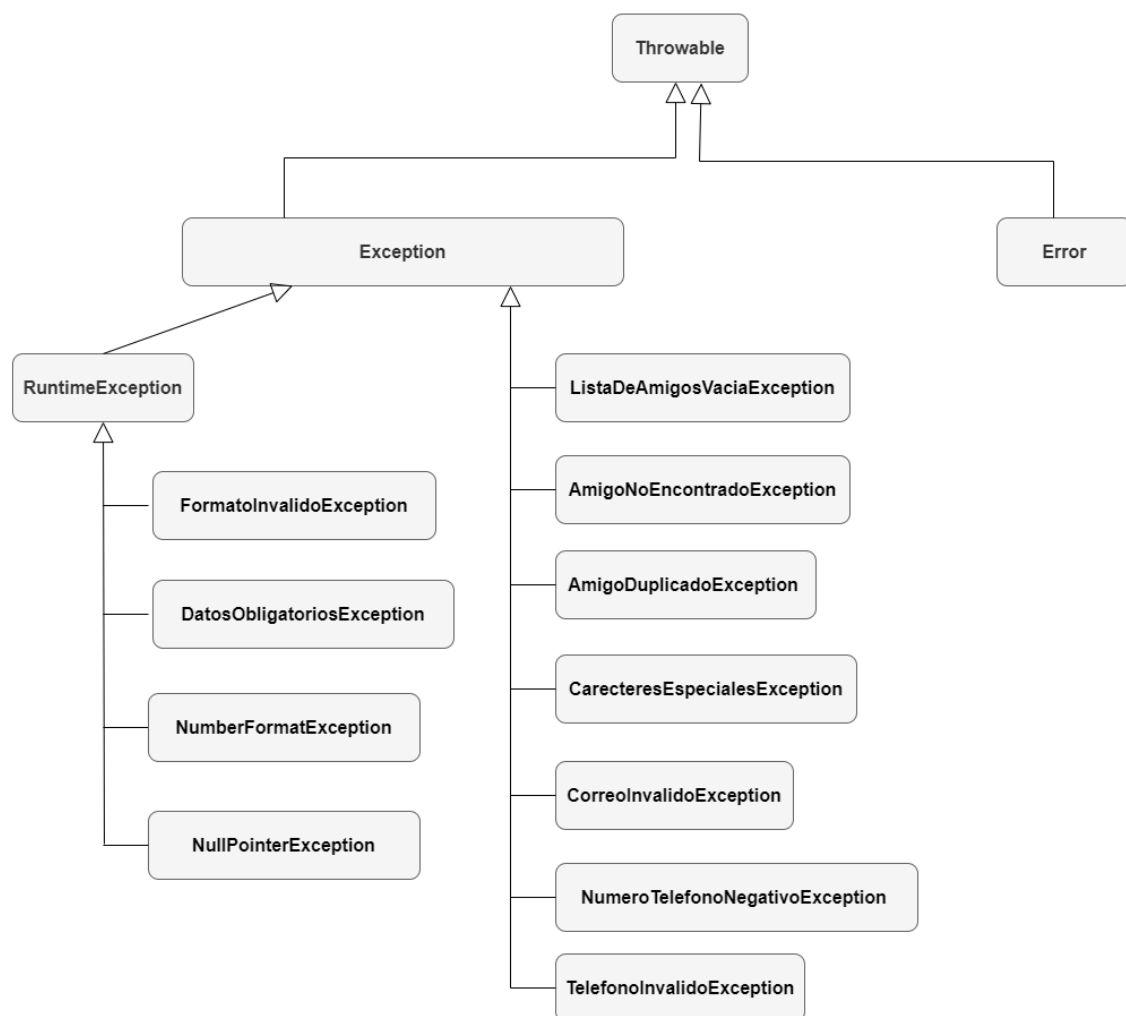
Diagrama de clases



2. Listado de excepciones propias implementadas.

- DatosObligatoriosException** : Se lanzará cuando algunos de los campos este vacío.
- CorreoInvalidoException**: Se lanzará cuando el correo no tenga un “@”
- TelefonoInvalidoException** : Se lanzará cuando el numero de telefono no empiece “606” o “30”
- AmigoDuplicadoException** : Se lanzará cuando intenten registrar a otro amigo con el mismo correo.
- ListaDeAmigosVacíaException**: Se lanzará cuando la lista de amigos esté vacía.

- NumeroTelefonoNegativoException : Se lanzará cuando el usuario ingrese un numero negativo.
- AmigoNoEncontradoException: Se lanzará cuando no se encuentre el amigo que el usuario intenta buscar.
- FormatoInvalidoException: Cuando el usuario tenga una mezcla de numeros y letras, donde solo son numeros o letras.
- CarecteresEspecialesException: Cuando el usuario intenta registrar solo caracteres especiales.
- NumberFormatException: Ocurre cuando se intenta convertir una cadena de texto en un número y la cadena no tiene un formato válido para esa conversión.
- NullPointerException: Ocurre cuando se intenta acceder a un objeto que es null o tiene ese valor.



3. Implementación de la aplicación y presentación final.

Enlace Git: <https://github.com/heily0118/DirectorioAmistadesApp.git>