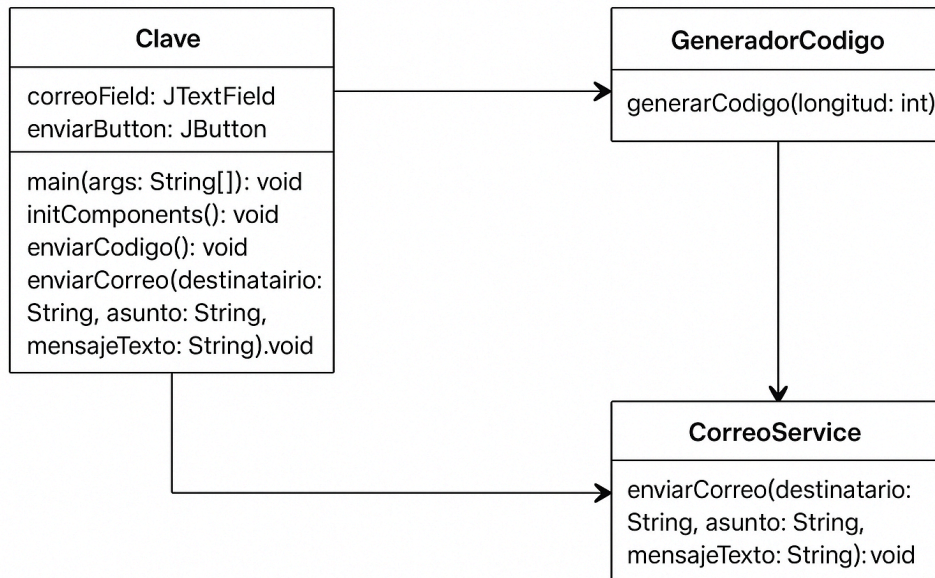


# DIAGRAMA DE CLASES



## 1. Clase Clase

Esta es la clase principal del programa. Representa la ventana Swing y coordina todo el flujo:

### ATRIBUTOS

- **correoField** : `JTextField`  
Campo donde el usuario escribe su correo electrónico.
- **enviarButton** : `JButton`  
Botón que el usuario presiona para generar y enviar el código.

### MÉTODOS

- **main(args: String[])** : `void`  
Inicia la aplicación. Solo crea la ventana y la hace visible.
- **initComponents()** : `void`  
Configura la interfaz gráfica: campos, botones, paneles, listeners.
- **enviarCodigo()** : `void`  
Método activado cuando se presiona el botón.
  - Obtiene el correo
  - Genera un código
  - Envía el correo usando los otros módulos
- **enviarCorreo(destinatario, asunto, mensajeTexto)** : `void`  
Aquí estuvo originalmente la lógica SMTP.  
En el diagrama aparece separado en `CorreoService` porque esa responsabilidad debe ir aparte (principio SRP).

## RELACIONES

- Flecha hacia `GeneradorCodigo`  
Porque usa su método `generarCodigo()`.
- Flecha hacia `CorreoService`  
Porque usa su método para enviar correos.

Responsabilidad: Coordinar el proceso completo y manejar la interfaz gráfica (GUI).

## 2. Clase `GeneradorCodigo`

Esta clase es completamente independiente y se dedica solo a crear contraseñas o códigos aleatorios.

### MÉTODOS

- **`generarCodigo(longitud: int)`**  
Devuelve un string aleatorio con letras y números según la longitud requerida.

Responsabilidad: Generar códigos — nada más.

## 3. Clase `CorreoService`

Encapsula toda la lógica relacionada con el envío de correos.

### MÉTODOS

- **`enviarCorreo(destinatario, asunto, mensajeTexto)` : void**  
Contiene la configuración SMTP, autenticación y envío del mensaje.

Antes estaba dentro de la clase `Clave`, pero queda mejor separarlo para:

- Mejor mantenimiento
- Poder reutilizarlo
- Facilitar pruebas (unit testing)

Responsabilidad: Enviar correos electrónicos de manera independiente del resto del programa.