

## Contents

<b>1 Documentación da Axenda de Contactos</b>	<b>2</b>
1.1 Introducción . . . . .	2
1.2 Estrutura do Programa . . . . .	2
1.2.1 As 3 Capas Principais . . . . .	2
1.3 Funcionalidades Principais . . . . .	2
1.3.1 1. Listado de Contactos . . . . .	2
1.3.2 2. Alta de Contactos . . . . .	3
1.3.3 3. Busca de Contactos . . . . .	3
1.4 Como Funciona o Programa (Fluxo Principal) . . . . .	4
1.4.1 O Bucle Principal . . . . .	4
1.5 Validación de Datos . . . . .	5
1.5.1 Que se valida automáticamente? . . . . .	5
1.6 Boas Prácticas para Principiantes . . . . .	5
1.6.1 1. Xestión de Cancelacións . . . . .	5
1.6.2 2. Confirmacións Importantes . . . . .	5
1.6.3 3. Manexo de Erros . . . . .	5
1.6.4 4. Non Modifiques Directamente . . . . .	6
1.7 Erros Comúns e Solucións . . . . .	6
1.7.1 Erro 1: Axenda Chea . . . . .	6
1.7.2 Erro 2: Contacto Duplicado . . . . .	6
1.7.3 Erro 3: Datos Incorrectos . . . . .	6
1.7.4 Erro 4: Cancelacións Non Xestionadas . . . . .	6
1.8 Exemplo Completo de Uso . . . . .	7
1.8.1 Secuencia Típica para un Novo Usuario . . . . .	7
1.9 Por que é un Bo Exemplo de Programación? . . . . .	8
1.9.1 1. Separa Responsabilidades . . . . .	8
1.9.2 2. Valida Datos Correctamente . . . . .	8
1.9.3 3. Boa Experiencia de Usuario . . . . .	8
1.9.4 4. Código Límpio e Mantible . . . . .	8
1.9.5 5. Arquitectura Escalable . . . . .	8
1.10 Próximos Pasos para Aprender . . . . .	8
1.10.1 1. Practica Coa Axenda Actual . . . . .	8
1.10.2 2. Mellora a Aplicación (Reto!) . . . . .	9
1.10.3 3. Profundiza nos Conceptos . . . . .	9
1.10.4 4. Crea Variacións . . . . .	9
1.11 Resumo de Conceptos Clave . . . . .	9
1.12 Consellos Finais . . . . .	9

# 1 Documentación da Axenda de Contactos

## 1.1 Introducción

Esta aplicación é unha **axenda de contactos completa** que permite: - Gardar contactos con nome, teléfono, email e descripción - Buscar contactos por diferentes criterios - Modificar e eliminar contactos existentes - Validar que os datos sexan correctos antes de gardalos

É un exemplo perfecto de como organizar un programa en **capas** (presentación, negocio e datos) e como usar **boas prácticas** de programación.

---

## 1.2 Estrutura do Programa

### 1.2.1 As 3 Capas Principais

```
// 1. CAPA DE DATOS (almacenamiento)
ContactosDAO axenda = new ContactosDAO(25);

// 2. CAPA DE NEGOCIO (lógica)
Contacto contacto = new Contacto(persoa, "612345678", "email@email.com", "Amigo");

// 3. CAPA DE PRESENTACIÓN (interface de usuario)
Forms.contactoForm();
```

#### Para principiantes:

Cada capa ten unha responsabilidade clara: - **Datos**: Gardar e recuperar información - **Negocio**: Definir que é un contacto, validación de regras - **Presentación**: Interactuar co usuario

---

## 1.3 Funcionalidades Principais

### 1.3.1 1. Listado de Contactos

**Que fai:** Amosa todos os contactos gardados  
**Como acceder:** Opción 1 do menú principal

#### Exemplo de uso:

```
Axenda de Contactos
-----
1.- Listado de Contactos
2.- Alta de Contactos
3.- Busca de Contactos
4.- Saír
Eixe opción: 1
```

Listado de Contactos

---

- 1) García, Ana: 612345678 (ana@email.com) - Colega
- 2) Rodríguez, Carlos: 698765432 (carlos@email.com) - Traballo

Para continuar preme ENTER...

---

### 1.3.2 2. Alta de Contactos

**Que fai:** Crea un novo contacto validando todos os datos

**Como acceder:** Opción 2 do menú principal

**Pasos do proceso:** 1. Introduce os datos da persoa (NIF, nome, apelidos) 2. Introduce o teléfono e email 3. Engade unha descripción 4. Confirma antes de gardar

**Exemplo:**

Alta de Contacto

---

Nova Persoa

---

Introduce Nif: (\* para cancelar): 12345678Z

Introduce Nome: (\* para cancelar): Ana

Introduce Apellidos: (\* para cancelar): García

Telefono: (\* para cancelar): 612345678

Email: (\* para cancelar): ana@email.com

Descripción: (\* para cancelar): Colega

Está vostede seguro? (S/N): S

Contacto gardado. Para seguir preme Enter.

---

### 1.3.3 3. Busca de Contactos

**Que fai:** Busca contactos por diferentes criterios e permite modificalos ou borrarlos

**Como acceder:** Opción 3 do menú principal

**Criterios de busca:** - **N:** Por NIF da persoa - **E:** Por email - **T:** Por teléfono - **D:** Por descripción (palabra chave)

**Exemplo de busca:**

Por (N)if, (E)mail, (T)eléfono ou (D)escrición? (NETD): N

Nif: (\* para cancelar): 12345678Z

```

Resultados (Eixe un para modificar ou borrar)
-----
1.- 1)García, Ana: 612345678 (ana@email.com) - Colega
2.- Sair
Eixe opción: 1

Edición de Contacto
-----
1)García, Ana: 612345678 (ana@email.com) - Colega
(M)odificar, (B)orrar (MB): M

Edición de Contacto
-----
Datos Persoais
-----
Nif: 12345678Z
Nome: Ana
Apelidos: García
Telefono: (612345678): 611223344
Email: (ana@email.com): ana.novo@email.com
Descripción: (Colega): Mellor amiga

Para continuar preme ENTER...

```

## 1.4 Como Funciona o Programa (Fluxo Principal)

### 1.4.1 O Bucle Principal

```

public static void main(String[] args) {
    int op = 0;
    do {
        op = Input.menu("Axenda de Contactos", menu);
        switch (op) {
            case 1: listaContacto(); break;
            case 2: altaContacto(); break;
            case 3: buscaContacto(); break;
        }
    } while (op != 4); // 4 = Sair
}

```

**Explicación para principiantes:**

1. Amosa o menú con Input.menu()
2. Espera a elección do usuario
3. Executa a acción correspondente: - 1 → Amosa todos os contactos - 2 → Crea un novo contacto - 3 → Busca contactos
4. Repite ata que o usuario escolla

“Saír” (4)

---

## 1.5 Validación de Datos

### 1.5.1 Que se valida automáticamente?

**NIF:** 8 números + 1 letra correcta

**Teléfono:** Exactamente 9 díxitos

**Email:** Formato correcto con @

**Campos obligatorios:** Nome e apelidos non poden estar baleiros

**Exemplo de validación:**

```
Introduce Nif: (* para cancelar): 1234567
Erro: NIF debe ter 8 números e 1 letra
Introduce Nif: (* para cancelar): 12345678A
Erro: DNI non valido. A letra non coincide.
Introduce Nif: (* para cancelar): 12345678Z  Correcto!
```

---

## 1.6 Boas Prácticas para Principiantes

### 1.6.1 1. Xestión de Cancelacións

O usuario pode **cancelar** en calquera momento premendo \*:

```
try {
    String nome = Input.readText("Nome:");
} catch (CancelException e) {
    System.out.println("Operación cancelada");
    return; // Volve ao menú
}
```

### 1.6.2 2. Confirmacións Importantes

Antes de accións irreversibles (como borrar), pide confirmación:

```
if (Input.areYouSure()) {
    axenda.eliminarContacto(numero);
}
```

### 1.6.3 3. Manexo de Erros

Captura e amosa errores de forma amigable:

```
try {
    axenda.guardarContacto(contacto);
} catch (StorageFull e) {
```

```

        System.out.println("Erro: Axenda chea!");
    } catch (DuplicateElement e) {
        System.out.println("Erro: Contacto duplicado!");
    }
}

```

#### 1.6.4 4. Non Modifiques Directamente

Nunca accedas directamente aos atributos:

```

// MAL!
contacto.telefono = "612345678";

// BEN!
contacto.setTelefono("612345678"); // Valida automaticamente

```

---

### 1.7 Erros Comúns e Solucións

#### 1.7.1 Erro 1: Axenda Chea

**Síntoma:** “Erro: teléfono non válido” ou similar

**Causa:** A axenda só ten espazo para 25 contactos

**Solución:** Elimina contactos antigos ou crea unha axenda máis grande

#### 1.7.2 Erro 2: Contacto Duplicado

**Síntoma:** “Erro: Elemento Duplicado”

**Causa:** Un contacto co mesmo NIF e descripción xa existe

**Solución:** Modifica a descripción ou actualiza o contacto existente

#### 1.7.3 Erro 3: Datos Incorrectos

**Síntoma:** O programa non acepta o NIF, teléfono ou email

**Causa:** Formato incorrecto

**Solución:** Sigue os formatos: - **NIF:** 8 números + 1 letra (ex: 12345678Z) -

**Teléfono:** 9 díxitos (ex: 612345678)

- **Email:** usuario@dominio.com

#### 1.7.4 Erro 4: Cancelacións Non Xestionadas

**Síntoma:** O programa pecha cando se preme \*

**Causa:** Falta capturar CancelException

**Solución:** Usa sempre try-catch:

```

try {
    // código que pode cancelarse
} catch (CancelException e) {
}

```

```
    System.out.println("Operación cancelada");
}
```

---

## 1.8 Exemplo Completo de Uso

### 1.8.1 Secuencia Típica para un Novo Usuario

Axenda de Contactos

- 1.- Listado de Contactos  
2.- Alta de Contactos  
3.- Busca de Contactos  
4.- Saír

Elixe opción: 2

Alta de Contacto

-----  
Nova Persoa

-----  
Introduce Nif: (\* para cancelar): 12345678Z  
Introduce Nome: (\* para cancelar): Ana  
Introduce Apellidos: (\* para cancelar): García  
Telefono: (\* para cancelar): 612345678  
Email: (\* para cancelar): ana@email.com  
Descripción: (\* para cancelar): Colega

Está vostede seguro? (S/N): S

Contacto gardado. Para seguir preme Enter.

Axenda de Contactos

- 1.- Listado de Contactos  
2.- Alta de Contactos  
3.- Busca de Contactos  
4.- Saír

Elixe opción: 1

Listado de Contactos

-----  
1)García, Ana: 612345678 (ana@email.com) - Colega

Para continuar preme ENTER...

Axenda de Contactos

---

```
1.- Listado de Contactos
2.- Alta de Contactos
3.- Busca de Contactos
4.- Sair
```

Eixe opción: 4

---

## 1.9 Por que é un Bo Exemplo de Programación?

### 1.9.1 1. Separa Responsabilidades

- Cada clase ten unha única función clara
- Fácil de modificar unha parte sen afectar ás outras

### 1.9.2 2. Valida Datos Correctamente

- Non acepta información incorrecta
- Prevén erros antes de que ocorran

### 1.9.3 3. Boa Experiencia de Usuario

- Mensaxes claras e en galego
- Opcións de cancelación en calquera momento
- Confirmacións para accións importantes

### 1.9.4 4. Código Límpio e Mantible

- Comentarios explicativos
- Métodos curtos e con nomes claros
- Estrutura lóxica e organizada

### 1.9.5 5. Arquitectura Escalable

- Se queremos cambiar a almacenamiento (de array a base de datos), só temos que modificar `ContactosDAO`
  - Se queremos cambiar a interface de usuario (de consola a gráfica), só temos que modificar `Forms` e `Input`
- 

## 1.10 Próximos Pasos para Aprender

### 1.10.1 1. Practica Coa Axenda Actual

- Crea 5 contactos diferentes
- Proba buscar por cada criterio
- Intenta modificar e eliminar contactos

- Proba introducir datos incorrectos para ver os erros

#### 1.10.2 2. Mellora a Aplicación (Reto!)

- Aumenta a capacidade da axenda a 50 contactos
- Engade unha opción para borrar todos os contactos
- Amosa o número total de contactos no menú principal
- Ordena os contactos alfabeticamente ao amosalos

#### 1.10.3 3. Profundiza nos Conceptos

- **Estuda as clases DAO:** Como funcionan ContactosDAO e DuplicateElement
- **Analiza as excepcións:** Como se xestionan StorageFull e CancelException
- **Entende a validación:** Como funcionan as expresións regulares en Input

#### 1.10.4 4. Crea Variacións

- **Axenda familiar:** Só garda nome e teléfono
  - **Axenda profesional:** Engade campos como empresa e cargo
  - **Axenda por categorías:** Permite agrupar contactos por tipo (amigos, traballo, etc.)
- 

### 1.11 Resumo de Conceptos Clave

Concepto	Explicación Simple
<b>DAO</b>	Clase que xestiona o almacenamento de datos
<b>Capas</b>	Separa o programa en partes especializadas
<b>Validación</b>	Comproba que os datos sexan correctos antes de usalos
<b>Excepcións</b>	Xestión de errores de forma ordenada
<b>Inmutabilidade</b>	Algúns datos non poden cambiar unha vez creados
<b>Encapsulamento</b>	Os datos están protexidos e só se accede mediante métodos

---

### 1.12 Consellos Finais

**Comeza por partes pequenas:** Entende primeiro como crear un contacto antes de intentar entender todo o programa

**Proba con exemplos reais:** Crea contactos con datos dunha persoa real para facer máis significativo o aprendizaxe

**Non temas os errores:** Os mensaxes de erro son axudas para entender que falla,

non fracasos

**Pregunta e experimenta:** Cambia números, nomes e proba diferentes opcións para ver que ocorre

**Divide problemas complexos:** Se algo non funciona, sepárao en partes máis pequenas até atopar o erro

Esta aplicación é un excelente punto de partida para entender como construír programas reais en Java. Cando dominres o seu funcionamento, estarás preparado para crear aplicacíons más complexas e útiles!