

## Práctica 4 (II) Gestor de correo fdimail

Grado en Ingeniería Informática  
Grado en Ingeniería del Software  
Grado en Ingeniería de Computadores

Virginia Francisco Gilmartín  
Raquel Hervás Ballesteros  
Facultad de Informática  
Universidad Complutense



### Orden de implementación

Entrega: 17/05 (23:55)

Módulos Correo y ListaCorreos

✓ 17/04

Módulo ListaRegistros

✓ 24/04

Módulo Usuario y ListaUsuarios

✓ 01/05

Módulo Gestor y Principal

✓ 08/05

Opcionales

✓ 15/05



# Gestor

---

- ✓ Tipos de datos:
  - tGestor:
    - Dominio (string), ej. “fdimail.com”
    - Usuarios → tListaUsuarios
    - Correos → tListaCorreos
    - Usuario activo (int) → Posición en la lista de usuarios del usuario activo en ese momento
- ✓ Subprogramas:
  - void inicializar(tGestor &gestor, string dominio):
    - ✓ Asigna a gestor el dominio
    - ✓ Inicializa la lista de usuarios y de correos
    - ✓ Pone como usuario activo -1



# Gestor

---

- bool arrancar(tGestor &gestor, string dominio):
  - ✓ Inicializamos el gestor
  - ✓ Cargamos la lista de usuarios
  - ✓ Cargamos la lista de correos
  - ✓ Devolvemos si todo ha ido bien
- void apagar(const tGestor &gestor):
  - ✓ Guardamos la lista de usuarios
  - ✓ Guardamos la lista de correos



# Gestor

---

- `bool crearCuenta(tGestor &gestor):`
  1. Solicitamos datos de acceso (id, contraseña)
  2. Buscamos el usuario (id + "@" + dominio) en la lista de usuarios
  3. Si ya se encuentra en la lista mostramos un mensaje de error
  4. Si no se encuentra en la lista:
    1. Creamos un nuevo usuario con el id y la contraseña introducidas
    2. Añadimos el nuevo usuario a la lista de usuarios del gestor
    3. Si se ha añadido correctamente buscamos el usuario en la lista para obtener la posición que ocupa
    4. Ponemos la posición del usuario como usuario activo del gestor



# Gestor

---

- `bool iniciarSesion(tGestor &gestor):`
  1. Solicitamos datos de acceso (id, contraseña)
  2. Buscamos el usuario (id + "@" + dominio) en la lista de usuarios
  3. Si no se encuentra en la lista mostramos un mensaje de error y pondremos como usuario activo -1
  4. Si se encuentra en la lista:
    1. Validamos la contraseña
    2. Si la contraseña es válida ponemos como usuario activo la posición que ocupa el usuario en la lista de usuarios del gestor
    3. Si la contraseña no es válida ponemos -1 como usuario activo del gestor



# Gestor

– int menuBandeja(const tGestor &gestor, bool entrada):

```
Correo de ginebra@fdinail.com
Bandeja de entrada-----
L N      EMISOR          ASUNTO                      FECHA
-----
* 1 - lancelot@fdinail.com Besos y abrazos          2015/3/23
* 2 - arturo@fdinail.com  ¡Mesa en oferta!        2015/3/23
* 3 - arturo@fdinail.com  Petición a Lancelot     2015/3/23
-----
Elija una opcion:
1- Leer correo
2- Enviar correo
3- Borrar correo
4- Ver bandeja de salida
5- Lectura rapida de correos sin leer
0- Cerrar sesion
-----
Introduzca una opcion:
```

- ✓ Mostramos (al revés) los correos recibidos o enviados (según entrada) del usuario activo
- ✓ Mostramos las opciones del menú
- ✓ Devolvemos la opción introducida por el usuario (siempre una opción correcta 0-5)



# Gestor

– void ejecutarMenuBandeja(tGestor &gestor, int opcion, bool &entrada):

✓ Realiza la opción elegida por el usuario:

1. Leer correo
2. Enviar correo
3. Borrar correo
4. Ver bandeja contraria
5. Lectura rápida de correos sin leer



*Excepto la opción 4, cada una tiene un subprograma asociado*



# Gestor

- void leerCorreo (tGestor &gestor, tListaRegCorreo &listaReg)
  - ✓ Solicitamos la posición del correo que desea leer
  - ✓ Transformamos la posición a la posición correcta



*Los correos se muestran al revés*

- ✓ Ponemos el correo como leído
- ✓ Buscamos el correo en la lista de correos del gestor
- ✓ Mostramos el correo con sus opciones



# Gestor

- void menuLeerCorreo (tGestor &gestor, const tCorreo &correoOriginal)

```
De: arturo@fdinail.com 2015/3/23 (20:28:6)
Para: lancelet@fdinail.com; ginebra@fdinail.com
Asunto: ¡Mesa en oferta!

Hay una mesa muy chula de oferta en Ikea!
¿La pillamos?

Ciao,
Arturo

-----
Elija una opcion:
1- Contestar correo
0- Volver a la bandeja
-----
Introduzca una opcion:
```

- ✓ Mostramos el correo por pantalla
- ✓ Mostramos las opciones:
  - ✓ Contestar:
    - ✓ Creamos un correo contestación
    - ✓ Lo enviamos
  - ✓ Volver: Mostramos mensaje de despedida



## Gestor

---

- void enviarCorreo (tGestor &gestor, const tCorreo &correo)
  - ✓ Insertamos el correo en la lista de correos del gestor
  - ✓ Creamos un registro para el usuario activo y lo insertamos en su lista de enviados
  - ✓ Creamos un registro para el receptor y lo insertamos en su lista de recibidos



## Gestor

---

- void borrarCorreo (tGestor &gestor, tListaRegCorreo listaReg)
  - ✓ Solicitamos la posición del correo a borrar al usuario
  - ✓ Transformamos la posición introducida a la posición correcta de la lista
  - ✓ Obtenemos el identificador del correo a borrar
  - ✓ Borramos el correo




*Los correos se muestran al revés*



# Gestor

– void lecturaRapida (tGestor &gestor, tListaRegCorreo &listaReg)

1. Vamos recorriendo todos los registros de listaReg:
  - ✓ Si el registro esta leído no hacemos nada
  - ✓ Si el registro no está leído:
    - a) Lo marcamos como leído
    - b) Buscamos el correo que corresponde al registro en la lista de correos del gestor
    - c) Añadimos el correo a una lista auxiliar de correos
2. Ordenamos la lista de correos auxiliar por asunto y fecha

 ordenar\_AF(...) del módulo listaCorreos

3. Mostramos la lista de correos auxiliar

 aCadena(...) del módulo listaCorreos



# Principal

- ✓ Arrancamos el gestor
- ✓ Mostramos el menú principal de la aplicación

```
Elija la opcion deseada:  
1- Acceder a mi cuenta de correo  
2- Crear cuenta de correo  
  
0- Salir
```

- ✓ Según la opción introducida:
  - 1:
    - ✓ Iniciamos sesión
    - ✓ Mostramos el menú del usuario
  - 2:
    - ✓ Creamos un nueva cuenta
    - ✓ Mostramos el menú del usuario
  - 0: Cerramos
    - ✓ Apagamos el gestor



# Depuración

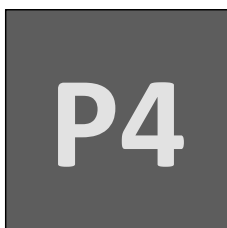
---

- Probar, probar y probar
- Hay que probar todos los casos que se os ocurran, sobre todo aquellos casos que son propensos a error:
  - Arrancar sin ficheros, con ficheros vacíos, ficheros con más elementos de los permitidos, con un solo elemento...
  - Iniciar sesión con usuario que no existe, con usuario que existe, con contraseña correcta, incorrecta...
  - Crear usuario con identificador que ya existe, con otro que no existe...
  - Enviar correos a usuarios inexistentes, a usuarios que existen, con cuerpo vacío...
  - Borrar un correo enviado y comprobar que sigue estando en la bandeja de entrada del remitente
  - Enviar más correos que los que caben en una bandeja de salida, en una bandeja de entrada...
  - Salir del gestor y volver a entrar para comprobar que todo se guardo bien
  - ...



## Fundamentos de Programación

---



# Práctica 4 (II) Gestor de correo fdimail

Grado en Ingeniería Informática  
Grado en Ingeniería del Software  
Grado en Ingeniería de Computadores

Virginia Francisco Gilmartín  
Raquel Hervás Ballesteros  
Facultad de Informática  
Universidad Complutense

