



UNIVERSIDAD
DE MÁLAGA

Departamento de Lenguajes
y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

Gestión de la Información

Prueba 1

Curso 2014/2015

En este ejercicio se va a desarrollar una agenda electrónica muy simplificada, con la que gestionaremos una tabla de contactos (`tContactos`) en la base de datos `P1_1415`.

tContacto		
	Nombre de columna	Tipo de datos
	nombre	varchar(50)
	apellido	varchar(50)
	email	varchar(50)
🔑	telefono	varchar(50)

- 1) La clase `Error` es una excepción no comprobada que nos permite tratar las diferentes situaciones excepcionales (clase suministrada).
- 2) El enumerado `Campo` se utiliza para identificar los posibles campos que se mantendrán para cada uno de los contactos. Los valores para este enumerado son `NOMBRE`, `APELLIDO`, `EMAIL` y `TELEFONO` (clase suministrada).
- 3) (**2 ptos.**) La clase `BD` se utiliza para simplificar las operaciones de acceso a la base de datos (esqueleto de la clase suministrado).
- 4) (**5 ptos.**) La clase `Contacto` (esqueleto de la clase suministrado) debe mapear la información sobre un contacto en la base de datos. En concreto, tendrá un valor de tipo `String` para cada uno de los campos de un contacto. Cualquiera de los campos puede ser la cadena vacía, pero no `null`. También dispondrá del constructor y los métodos necesarios para:
 - a. Construir un objeto de la clase a partir del nombre, apellido, email y teléfono de un contacto. Al ser un objeto nuevo lo insertará en la tabla. Si cualquiera de los argumentos es `null` se lanzará una excepción `Error`.
 - b. Construir un objeto de la clase contacto con el teléfono como argumento que inicializa cada uno de los campos del contacto obteniendo sus valores de la tabla.
 - c. Método estático `List<Contacto> ListaContactos()` que devuelve una lista con todos los contactos existentes en la tabla.
 - d. Método `void BorraContacto()` que borra el contacto actual de la tabla y deja todos sus atributos a `null`.
 - e. Métodos `String getValor(Campo campo)` y `void setValor(Campo campo, String valor)` para, respectivamente, consultar y modificar cada uno de los campos de un contacto. Si el valor a asignar a un campo fuese `null` se lanzaría una excepción `Error`.

- f. La representación como cadena de caracteres (`String toString()`) de un contacto con valores "nombre", "apellido", "email" y "teléfono" será `"Contacto(nombre, apellido, teléfono, email)"`.
- g. Dos objetos de la clase `Contacto` son iguales si coinciden los valores de todos sus campos.
- h. El orden natural de la clase `Contacto` será el orden lexicográfico de sus apellidos. A igualdad de apellidos se ordenarán por nombre, a igualdad de nombres por teléfonos, y a igualdad de teléfonos por email.
- 5) La clase `Agenda` (clase suministrada) gestionará una agenda de contactos. La clase dispondrá del constructor y los métodos necesarios para:
- Construir un objeto de la clase inicialmente sin contactos con un constructor sin argumentos.
 - Construir un objeto de la clase y cargar en la base de datos los contactos que existen en un fichero. El formato de los datos es el que aparece en el fichero `contactos.txt`. En él, cada uno de los contactos aparece con el formato

```
<apellido>, <nombre>#<número de teléfono>#<email>
```
 - La representación de la agenda como cadena de caracteres consiste en una cadena con la forma

```
Contacto(...)
...
Contacto(...)
```
- 6) (**3 ptos.**) Deseamos por último desarrollar una GUI para nuestra agenda. Dada las clases desarrolladas en los apartados anteriores, y dadas la interfaz `AgendaVista` y la clase `AgendaPanel` proporcionadas, completa la clase `AgendaCtrl` para que controle e interactúe con la agenda y la vista. Los nombres de los ficheros serán proporcionados como argumentos del método `main`, de forma que al crear la agenda sean cargados los datos disponibles (utiliza la clase `MainGUI` proporcionada).

MiniAgenda

Kent, Clark#000000000#superman@super.mail
Parker, Peter#888888888#spiderman@super.mail
Rogers, Steven#999999999#captainamerica@super.mail
Stark, Tony#777777777#ironman@super.mail
Wayne, Bruce#333333333#batman@super.mail

Apellido	<input type="text"/>	BORRA
Nombre	<input type="text"/>	BUSCA
Teléfono	<input type="text"/>	AÑADE
E-mail	<input type="text"/>	RESETEA

Agenda creada

- Los campos de texto son los cuatro editables, y en cualquier momento podemos escribir en cualquiera de ellos.
- El área de texto no es editable, en él se muestra el estado de la agenda en cada momento. Tras las operaciones que modifican la agenda el texto será actualizado.
- Al pulsar el botón BORRA se elimina de la agenda el contacto con los datos de los campos de texto correspondientes. Si no hubiera un contacto con estos datos no se borrará nada. Los campos de texto son reseteados después de eliminar el contacto.
- Al pulsar el botón BUSCA se completan los datos de un contacto cuyos datos se correspondan con el teléfono que se haya introducido. Si no se encontrara ningún contacto satisfaciendo esta condición se dará un mensaje de error.
- Al pulsar el botón AÑADE se creará un nuevo contacto con los datos en los campos de texto correspondientes y se incluirá en la agenda. Si el contacto ya estuviera en la agenda se dará un mensaje de error.
- Al pulsar el botón RESETEA se dejan en blanco todos los campos de texto.
- Las operaciones que resultan en un comportamiento normal (sin mensaje de error) informarán de la realización de la operación.