

INSTRUCCIONES

1. Crear un proyecto nuevo de Kotlin con el nombre **actividadKotlin**, subirlo al repositorio de **GitHub** como proyecto público.
2. Clonar el proyecto en el **IntelliJ IDEA** para poder desarrollar cada programa.
3. El proyecto estará formado por un total de 2 programas y cada programa se desarrollará en una **feature** distinta: el programa 1 en una *feature* y el programa 2 en otra *feature* diferente.
4. Desarrollo de los **programas**, cada uno en su *feature* correspondiente.
5. **Entrega:** se debe enviar el enlace al proyecto de GitHub y el archivo o archivos con el código de todo el proyecto comprimido desde la tarea habilitada para tal fin en el campus.

NOTA: ante cualquier duda o problema en la realización de alguna de las tareas descritas, se podrá solicitar la aclaración al alumno e incluso solicitar la realización de alguna modificación para asegurar su autoría.

Si se detecta que el alumnado copia, no podrá continuar con el desarrollo de la actividad y suspenderá automáticamente el primer parcial.

PROGRAMAS

PROGRAMA 1

Escribe un programa que gestione una agenda telefónica de contactos:

- La agenda está formada por contactos. El número máximo de contactos se podrá indicar, pero si no se indica nada, será de 10 contactos.
- Un contacto está definido por un nombre y un teléfono. Dos contactos serán iguales si lo son sus nombres.
- Se podrán realizar las siguientes operaciones:
 1. Añadir un contacto a la agenda: si no se pueden meter más contactos a la agenda se indicará por pantalla. No se pueden meter contactos que sean iguales.
 2. Listar todos los contactos de la agenda.
 3. Buscar un contacto por su nombre. Si lo encuentra muestra su teléfono y si no, se indica por pantalla.
 4. Comprobar si un contacto existe.
 5. Eliminar un contacto. Indicar por pantalla si se eliminó o no.

6. Indicar el número de contactos que se pueden añadir en la agenda, es decir, indicar el número de huecos libres.
7. Comprobar si la agenda está llena.
8. Salir

Crea un menú con las distintas opciones por consola que se repita hasta que el usuario seleccione la opción Salir.

Si se selecciona cualquier otro número, que no sea del 1 al 8 mostrar mensaje indicándolo y que se vuelva a mostrar el menú.

Además de que se muestre correctamente el resultado, se valorará la legibilidad, la utilización adecuada de estructuras de datos y la entrega de la actividad por GitHub.

La **salida esperada** es:

- Menú:

```
1. Añadir contacto
2. Listar contactos
3. Buscar contacto
4. Existe contacto
5. Eliminar contacto
6. Contactos disponibles
7. Agenda llena
8. Salir
Escribe una de las opciones
```

- Opción 1. Añadir contacto
 - Añadir un contacto que no existe:

```
Escribe una de las opciones
1
Escribe un nombre
Patricia
Escribe un telefono
985985985
Se ha añadido el contacto
```

- Añadir un contacto que ya existe:

```
Escribe una de las opciones
1
Escribe un nombre
Patricia
Escribe un telefono
985212223
El contacto con ese nombre ya existe
```

- Añadir un contacto cuando la agenda está llena:

```
Escribe una de las opciones
1
Escribe un nombre
Lucia
Escribe un telefono
123456789
La agenda esta llena, no se pueden meter mas contactos
```

- Opción 2. Listar contactos

- Si no hay contactos en la agenda:

```
Escribe una de las opciones
2
No hay contactos que mostrar
```

- Si hay contactos en la agenda:

```
Escribe una de las opciones
2
Nombre = Patricia, Teléfono = 985985985
Nombre = David, Teléfono = 123456789
```

- Opción 3. Buscar contacto

- Si no existe el contacto en la agenda:

```
Escribe una de las opciones
3
Escribe un nombre
Juan
No se ha encontrado el contacto
```

- Si existe el contacto en la agenda:

```
Escribe una de las opciones
3
Escribe un nombre
Patricia
Su telefono es 985985985
```

- Opción 4. Comprobar la existencia de un contacto

- El contacto existe:

```
Escribe una de las opciones
4
Escribe un nombre
David
Existe contacto
```

- El contacto no existe:

```
Escribe una de las opciones
4
Escribe un nombre
Patri
No existe contacto
```

- Opción 5. Eliminar un contacto

- Eliminar un contacto que no existe:

```
Escribe una de las opciones
5
Escribe un nombre
Patri
No se ha eliminado el contacto
```

- Eliminar un contacto que existe:

```
Escribe una de las opciones
5
Escribe un nombre
Patricia
Se ha eliminado el contacto
```

- Opción 6. Indicar el número de huecos libres en la agenda:

- La agenda no tiene huecos libres:

```
Escribe una de las opciones
6
Hay 0 contacto/s libre/s
```

- La agenda tiene huecos libres:

```
Escribe una de las opciones
6
Hay 1 contacto/s libre/s
```

- Opción 7. Comprobar si la agenda está llena

- La agenda está llena:

```
Escribe una de las opciones
7
La agenda está llena
```

- La agenda no está llena:

```
Escribe una de las opciones
7
Aun se pueden meter contactos
```

- Opción 8. Salir: se finaliza la ejecución del programa

```
Escribe una de las opciones
8

Process finished with exit code 0
```

PROGRAMA 2

Escribe un programa que muestre por consola los números de 1 a 100 (ambos incluidos y con un salto de línea entre cada impresión), sustituyendo los siguientes:

- Múltiplos de 3 por la palabra "tri"
- Múltiplos de 5 por la palabra "qui"
- Múltiplos de 3 y de 5 a la vez por la palabra "triqui"

Además de que se muestre correctamente el resultado, se valorará la legibilidad, la utilización adecuada de estructuras de datos y la entrega de la actividad por GitHub.

El **resultado esperado** al ejecutar el programa:

1	11	tri	31	41	tri	61	71	tri	91
2	tri	22	32	tri	52	62	tri	82	92
tri	13	23	tri	43	53	tri	73	83	tri
4	14	tri	34	44	tri	64	74	tri	94
qui	triqui	qui	qui	triqui	qui	qui	triqui	qui	qui
tri	16	26	tri	46	56	tri	76	86	tri
7	17	tri	37	47	tri	67	77	tri	97
8	tri	28	38	tri	58	68	tri	88	98
tri	19	29	tri	49	59	tri	79	89	tri
qui	qui	triqui	qui	qui	triqui	qui	qui	triqui	qui