

### **UF1. Projecte. Exercici d'integració**

#### **Aplicació per fer un sorteig. Fidelització de clients.**

##### ***Enunciat***

L'òptica del barri es trasllada, i vol donar a conèixer als clients la seva nova ubicació. Ha consultat a la nostra empresa com es pot gestionar a nivell tecnològic i quines accions de màrqueting li recomanem.

Es va plantejar la implantació d'un **CRM** (Customer Relationship Management), que és una eina per a la gestió de clients, però suposa una inversió massa gran ara i ens demanen alguna acció més simple i fàcil de posar en funcionament.

S'acorda llavors fer una acció inicial per aconseguir informar del trasllat i alhora avaluar com responen els clients, i segons com vagi, més endavant, invertir en un **CRM**.

Aquesta acció serà un sorteig entre tots els clients que facin una compra durant la primera setmana a la nova ubicació, d'aquesta manera es fomenta que els clients hi vagin i coneguin les noves instal·lacions.

Per donar suport a la recollida de dades, ens demanen una aplicació senzilla. Aquesta ha de permetre registrar les dades del client quan fa una compra. Més endavant, l'aplicació ens servirà per fer el sorteig entre tots els clients registrats.

En aquesta primera etapa, el programa ha de fer el següent:

- demanar les dades d'un client (id, edat, tipus de venda, import de la compra, telèfon de contacte)
- controlar la correcció de les dades entrades
- mostrar per pantalla un resum de les dades introduïdes

## Entrada de dades

### Tipus de venda

El camp *tipus de venda* és un valor sencer que cal guardar en una variable anomenada *tipus*, i indica quin tipus de venda s'ha fet.

Per entrar tipus, el programa haurà de donar les següents opcions:

*tipus de venda?:*

*venda lliure (0)*

*pensionista (1)*

*carnet jove (2)*

*soci (3)*

**: 0**

**L'usuari haurà d'introduir el codi del tipus de venda, segons indica el menú.**

### Control d'errors

El programa ha d'incorporar control d'errors. Les dades entrades han de tenir les següents característiques:

- id: sencer. Valors vàlids de 111 a 999 (ambdós inclosos).
- edat: sencer. Valors vàlids de 14 a 120 (ambdós inclosos).
- tipus de venda: sencer. Valors vàlids de 0 a 3 (ambdós inclosos).
- import de la compra: sencer. Valors vàlids de 0 a 1000 (ambdós inclosos).
- telefon: sencer. Valors vàlids de 111111111 a 999999999 (ambdós inclosos) i de 9 xifres (si n'introdueix un nombre de menys xifres o de més, no serà vàlid)

Si es produeix alguna errada en l'entrada de les dades, s'informa a l'usuari que hi ha un error en la dada introduïda i que torni a introduir un nou valor (dins del rang). Cal demanar fins a 3 cops una dada quan aquesta és incorrecta. Un cop esgotat aquest número d'intents (3 intents), cal finalitzar el programa sense fer res més (encara que només hagi introduït una de les dades).

*No es consideren correctes solucions que facin servir les instruccions `break`<sup>1</sup>, o `exit()` per sortir. Cal gestionar la sortida del programa amb estructures condicionals.*

---

<sup>1</sup> El `break` com a part d'una estructura `switch` sí que és correcte.



### Sortida de dades per pantalla

*Un cop entrada les dades, cal que es mostri un resum de totes les dades introduïdes.*

*Un exemple de sortida per pantalla podria ser el següent. En blau les dades que introdueix l'operador:*

Id?: 728

Edat?: 44

tipus de venda?:

venda lliure (0)

pensionista (1)

carnet jove (2)

soci (3)

: 0

import de la compra?: 24

telèfon de contacte?: 624084208

Id	edat	tipus	import	telèfon
728	44	lliure	24	624084208

*En aquest cas, l'usuari no ha introduït cap valor erroni*

*Per a elaborar aquest llistat, cal que seguiu les següents directives:*

- Cal definir els valors que no canvien com a constants: límits de valors vàlids, noms de tipus... etc
- Cal fer servir l'estructura switch per definir quin text mostrar segons sigui el valor del tipus de venda
- En finalitzar, cal mostrar totes les dades introduïdes en una sola fila.
- Cal validar que l'usuari no introdueixi tipus de valors diferents dels que s'esperen (evitar caràcters)

*A continuació teniu el joc de proves, per tal de validar que el programa funciona correctament i la rúbrica amb els criteris d'avaluació.*

### Lliurament

Crea una carpeta dins del teu repositori de GitHub i puja la carpeta del projecte i el document en format pdf de la rúbrica, amb els valors corresponents un cop revisat el teu projecte



### Joc de proves

id	Entrada	Sortida (què hauríeu d'obtenir) l·listat final										
1	Id: 9	Error en dades										
	Id: 10	Error en dades										
	Id: 100	Error en dades										
		Programa finalitzat per error en dades										
2	Id: 426	Error en dades										
	edat: 0	Error en dades										
	edat: 10	<table><tr><th>Id</th><th>edat</th><th>tipus</th><th>import</th><th>telèfon</th></tr><tr><td>426</td><td>14</td><td>c.jove</td><td>22</td><td>606034781</td></tr></table>	Id	edat	tipus	import	telèfon	426	14	c.jove	22	606034781
	Id	edat	tipus	import	telèfon							
	426	14	c.jove	22	606034781							
	edat: 14											
	tipus: 2											
	import: 22											
telf: 606034781												
3	Id: 111	Error en dades										
	edat: 13	<table><tr><th>Id</th><th>edat</th><th>tipus</th><th>import</th><th>telèfon</th></tr><tr><td>111</td><td>120</td><td>soci</td><td>780</td><td>460602759</td></tr></table>	Id	edat	tipus	import	telèfon	111	120	soci	780	460602759
	Id	edat	tipus	import	telèfon							
	111	120	soci	780	460602759							
	edat: 120											
	tipus: 3											
import: 780												
telf: 460602759												

### Criteris d'avaluació



Criteri	2 - Molt bé	1 - Suficient	0 - Insuficient	Dedicació (minuts)	Punts
<b>Robust</b>	El programa s'executa sense problemes i no mostra errors en execució	El programa s'executa, però en algun punt de l'execució falla i finalitza o mostra errors en execució	El programa no s'executa (dóna errors en compilació/execució)	3 hores	
<b>Claredat i ordre</b>	Es pot seguir fàcilment la seqüència de les instruccions. És endreçat i conserva l'estructura dels exemples vistos a classe i als materials.	El codi no està correctament estructurat: les estructures condicionals no discriminen casos sinó que repeteixen sovint comprovacions o el flux d'execució és més complex del necessari. A trossos les indentacions no són correctes.	No segueix en absolut l'estructura dels exemples. L'algoritme (idea que hi ha darrera del programa) és desendreçat. No està gens indentat.		
<b>Variables</b>	Els noms de les variables són prou autodescriptius. Els tipus de dades utilitzats són els adequats (els més eficients per a la tasca requerida). Les variables es declaren a principi del	Els noms de les variables són autodescriptius, però no tots els tipus de dades són correctes. No totes les variables estan declarades a principi del codi.	Els noms de les variables no són autodescriptius (no ens informen del seu contingut). Algun dels tipus no són correctes. Les variables no es declaren al principi del codi (es poden trobar declaracions al llarg del programa).		



	codi.				
<b>Constants</b>	Totes les dades que tenen un valor que no varia en tot el codi estan definides com a constants. Els noms de les constants és en MAJÚSCULES	Defineix com a constant algunes de les dades, però no d'altres que serveixen per marcar valors mínims i màxims i que no canvien en tot el codi.	No defineix constants. Els noms de les constants no estan en MAJÚSCULES.		
<b>Bucles</b>	Fa servir correctament els bucles, tal i requereix l'algorisme per a executar-se de la manera més eficient.	Fa servir els bucles però no controla correctament els valors límits, no incrementa correctament el comptador o la condició de sortida és incorrecta/inexistent..	No utilitza els bucles correctament.		
<b>Correcte</b>	Fa tot el que es demana. Passa correctament totes proves del joc de proves (debug).	Fa una part de les coses que es demanen, però no totes.	No fa el que es demana.		