



Prácticas de Programación – Práctica 1

Formato y fecha de entrega

La entrega debe hacerse en el apartado “Entregas y registro de EC” del aula de teoría, antes del día **3 d’abril de 2017 a les 23:55.**

Se debe entregar un fichero en formato ZIP con el nombre *ApellidosNombre_PP_PRACT1.zip* que contenga:

1. Fichero **PDF** en el cual expliquéis los problemas encontrados en la realización de los ejercicios. Se debe adjuntar una captura de pantalla del entorno final para cada ejercicio.
2. Carpeta / directorio “**UOCCarSharing**” con el espacio de trabajo utilizado. Antes de crear el fichero ZIP, eliminad los directorios temporales (desde CodeLite podéis ejecutar el comando clean)

Nota: Es posible que al hacer la entrega, el sistema os cambie el nombre del fichero. No os preocupéis por este hecho.

Presentación

Durante las diferentes prácticas desarrollaremos un único proyecto, del que iréis diseñando e implementando funcionalidades. En esta primera práctica implementaremos la navegación entre los menús de una aplicación para compartir coche entre usuarios. Posteriormente se extenderá esta funcionalidad para completar la aplicación.

Competencias

Transversales

- Capacidad de comunicación en lengua extranjera.

Específicas



- Capacidad de diseñar y construir aplicaciones informáticas mediante técnicas de desarrollo, integración y reutilización.
- Conocimientos básicos sobre el uso y la programación de ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en la ingeniería.

Objetivos

- Compilación de un proyecto de código.
- Comprensión del proyecto a desarrollar.
- Familiarización con las funcionalidades y el uso de código proporcionado por los profesores.
- Recordatorio de la sintaxis, el estilo y el entorno de programación en C.

Recursos

Per a realitzar esta pràctica dispondreu de los siguientes recursos:

Recursos Básicos

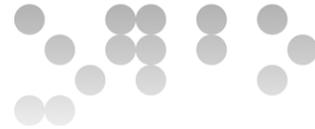
Laboratorio de Prácticas de Programación: Disponéis de un laboratorio asignado a la asignatura. En este laboratorio podéis resolver todas las dudas sobre la programación en lenguaje C, y la instalación y utilización del entorno de programación.

Material de Laboratorio: En el laboratorio, aparte del colaborador docente, disponéis de diferentes documentos que os ayudarán. En concreto, hay un *manual de C* y una *guía de programación en C*.

Transparencias de Síntesis: En el aula de teoría tenéis las transparencias de síntesis. Para esta práctica os será de especial interés la TS2 y la TS3.

Videos CodeLite: En el aula de laboratorio encontraréis una serie de videos sobre cómo utilizar el entorno de programación CodeLite..

Recursos Complementarios



Buscador web: La forma más rápida de obtener información actualizada sobre los parámetros, funcionamiento y ejemplos de utilización de cualquier método estándar de C (y en general de cualquier lenguaje), es mediante un buscador web.

Criterios de evaluación

Per a la valoració de los ejercicios se tendrá en cuenta:

- La adecuación de la respuesta a la pregunta formulada.
- Utilización correcta del lenguaje técnico y corrección en la redacción y ortografía.
- Claridad de la respuesta
- Completitud y nivel de detalle de la respuesta aportada.
- El código obtiene el resultado esperado dadas unas condiciones y datos de entrada diseñados para probar algunas situaciones de funcionamiento normal y otros casos especiales.
- El código entregado sigue la guía de estilo y las buenas prácticas de programación.
- Se separa correctamente la declaración e implementación de las acciones y funciones, utilizando los ficheros correctos.
- El código está correctamente comentado, valorando especialmente la utilización de comentarios en inglés.
- El grado de optimización en tiempo y recursos utilizados en la solución entregada.
- El código realizado es modular y estructurado, teniendo en cuenta la reutilización del código

Descripción de la Práctica a realizar

La práctica consiste en implementar una aplicación que permite compartir coche entre usuarios que quieren realizar un trayecto parecido. Así, los usuarios podrán publicar aquellos viajes que realicen de manera que, otros usuarios podrán visualizarlos y reservar una plaza en estos trayectos si así lo desean.

Durante las tres prácticas de la asignatura se irán completando diferentes partes del proyecto de manera incremental.

El proyecto consiste en la implementación de dos interfaces. En la primera que diseñaremos, los usuarios podrán publicar los trayectos, listarlos hacer reservas para otros trayectos. La segunda interfaz estará relacionada con la administración de los datos y permitirá tareas como la consulta de datos o borrar campos.

Nota: Los nombres de los tipos, de los atributos y de las operaciones se deben escribir en inglés. Es altamente recomendable redactar los comentarios también en inglés.



Nota: Hay que poner las declaraciones de las operaciones en los ficheros cabecera (*.h).

Ejercicio 1

El proyecto en esta primera práctica contiene dos partes claramente diferenciadas:

Aplicación CarShareApp: Es la aplicación que ejecutarán los usuarios de la aplicación y que servirá para publicar nuevos trayectos, listar los trayectos disponibles y reservar plazas.

Librería CarShareLib: Contendrá los tipos de datos, las definiciones de las estructuras para la gestión de la información y operaciones comunes a las aplicaciones.

Tarea 1

Cread un entorno de trabajo UOCCarSharing con la misma estructura que la proporcionada en el enunciado. Tened en cuenta los 2 proyectos que debe contener el entorno de trabajo así como las 2 vistas comentadas en la práctica 0: Debug y Release.

Indicad en el documento PDF una captura de pantalla con el entorno preparado.

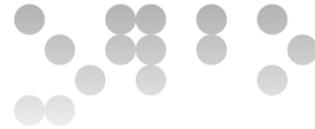
Tarea 2

En la librería *CarShareLib* existen los ficheros *helpers.c* y *helpers.h*. Estos ficheros contienen funciones auxiliares que nos permitirán realizar algunas operaciones comunes del programa.

Entre estas operaciones hay la función *helpers_clearScreen()* que contiene una llamada específica sólo para Sistemas Operativos Windows. El objetivo de esta función es la de limpiar la pantalla.

Adaptad esta función a vuestro Sistema Operativo y verificad que funciona correctamente. Encontraréis los comentarios de cómo adaptarla en el mismo código.

Nota: A lo largo de todas las prácticas del curso, no añadáis operaciones nuevas en estos ficheros.



Tarea 3

Probad de compilar y ejecutar el código proporcionado. Verificad que el siguiente menú os aparece:

```
Welcome to the UOC Car Sharing application!
-----
Library version: 1
Please, select a menu option:
1. Login
2. Register
0. Exit
```

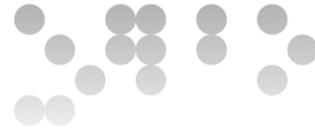
Ejercicio 2

Previo a poder identificarse, el usuario debe registrarse como usuario. Cuando un usuario se quiere registrar, el algoritmo llama a la función `appMenu_register` que pide los datos al usuario y los utiliza para rellenar la estructura que se guardará con toda la información.

Una vez tenemos el usuario registrado, éste podrá entrar en la aplicación utilizando sus credenciales. Para comprobar la certeza de los datos, se utiliza la función `appMenu_login` del fichero `appMenu.c`.

Tarea 1

Implementad la función `appMenu_register` para que pida los datos del usuario de la manera que se indica en la siguiente captura de pantalla.



```
Register
-----
Name :      Ricard
Surname :   Burriel
D.N.I. :    12345678A
Login :     ricard
Password :  ricBurr$04_
```

Los campos que pedirá para cada usuario son su nombre, apellidos, DNI, nombre de usuario y contraseña. (En la captura de pantalla, los campos de la izquierda los muestra la aplicación y los campos de la derecha los introduce el usuario).

Podéis utilizar la función `helpers_scanText` para obtener la cadena de caracteres introducida por el usuario.

Debéis guardar todos los datos introducidos en una estructura del tipo `user`. Una vez tenéis todos los datos, haced que la variable `users_firstUser` apunte a esta estructura.

Nota: Encontraréis la definición de la estructura con los campos que utiliza en el fichero cabecera `users.h`

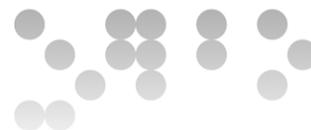
Nota: Recordad de utilizar la función `malloc` para reservar memoria para la estructura `user`.

Nota: Una vez hecho el `malloc`, llamad siempre a la función `users_initStruct` cuando creéis una nueva estructura del tipo `user`. Esta función simplemente inicializa todos los campos de la estructura a 0.

Tarea 2

Implementad la función `appMenu_login` para que pida el nombre de usuario y la contraseña del usuario.

```
Login
-----
Login :      ricard
Password :   ricBurr$04
```



Los campos a pedir son el nombre del usuario y la contraseña.

Igual que antes, podéis utilizar la función *helpers_scanText* para obtener la cadena de caracteres introducida por el usuario.

Una vez introducidos los datos, en esta primera práctica, la función debe devolver la variable global *users_firstUser*.

Nota: En posteriores prácticas se implementará la comprobación de los datos introducidos.

Ejercicio 3

Lo primero que hace la aplicación *CarShareApp* es mostrar el menú inicial para poder actuar en consecuencia.

La interacción con el usuario consiste en un bucle en el que se va procesando la entrada introducida por el usuario hasta que éste selecciona 0 para salir. Este bucle está implementado en la función *appMain_displayMainMenu* del fichero *appMain.c*

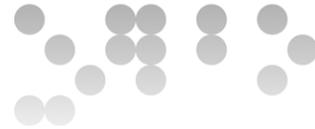
```
Welcome to the UOC Car Sharing application!
-----
Library version: 1
Please, select a menu option:
1. Login
2. Register
0. Exit
```

Tarea 1

Modificad la función *appMain_displayMainMenu* para que entre en un bucle que gestionará la interacción con el usuario.

Para mostrar la información al usuario y capturar la opción seleccionada, podéis llamar a la función *appMenu_mainMenu* del fichero *appMenu.c*.

La operación saldrá del bucle, con lo que retornará a la función principal, cuando el usuario apriete la tecla '0'.



Implementad la funcionalidad del bucle teniendo en cuenta las 2 opciones disponibles. En caso que el usuario seleccione '1', se llamará a la función de login, mientras que si selecciona la opción '2', se le permitirá registrarse en la aplicación.

Ejercicio 4

Una de las acciones principales que puede realizar un usuario es publicar un nuevo trayecto.

La función de publicar un nuevo trayecto está gestionada por el método `appMenu_publishNewTrip` del fichero `appMenu.c`.

Por otro lado, la función de mostrar toda la información sobre un trayecto en concreto está gestionada por el método `appMenu_displayTrip` del fichero `appMenu.c`.

Tarea 1

Implementad la función `appMenu_publishNewTrip` para que pida la información de un trayecto de la manera que se indica en la siguiente captura de pantalla.

```
Create a new trip
-----

Departure:      Barcelona
Arrival:        Girona
Number of available seats: 3
Departure date <yyyy/mm/dd hh:mm>: 2017/03/01 17:00_
```

Los campos que pedirá la aplicación para crear un nuevo trayecto son: punto de salida, punto de llegada, número de asientos disponibles y fecha y hora de salida. (En la captura de pantalla, los campos de la izquierda los muestra la aplicación y los campos de la derecha los introduce el usuario).

Podéis usar la función `helpers_scanText` para obtener la cadena de caracteres introducida por el usuario.

Debéis guardar todos los datos introducidos en una estructura del tipo `trip`. El apuntador a esta estructura será el valor que se devolverá en la función.



Nota: Encontraréis la definición de la estructura con los campos que utiliza en el fichero *trips.h*

Nota: Recordad de usar la función *malloc* para reservar memoria para la estructura *trip*.

Nota: Una vez hecho el *malloc*, llamad siempre a la función *trips_initStruct* cuando creéis una nueva estructura del tipo *trip*. Esta función simplemente inicializa todos los campos de la estructura a 0.

Tarea 2

Implementad la función *appMenu_displayTrip* para que muestre la información detallada del trayecto tal y como se muestra en la siguiente captura de pantalla.

```
Trip information:
-----
Departure:      Barcelona
Arrival:        Girona
Available seats: 3
DateTime:       2017/03/01 17:00
Press any key to continue.
```

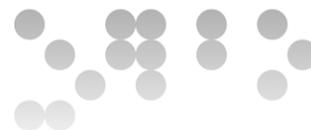
Los campos a mostrar son el punto de salida, el punto de llegada, el número de plazas disponibles y la fecha y hora de salida.

Nota: Tened en cuenta el formato con la fecha con 4 dígitos para el año y 2 dígitos para el resto de valores.

Ejercicio 5

Cuando el usuario se identifica en la aplicación, el programa muestra el menú de usuario para que éste pueda interactuar con el sistema.

Igual que antes, la interacción con el usuario consiste en un bucle en el que se va procesando la entrada introducida por el usuario hasta que éste selecciona la tecla 0 para salir. Este bucle está implementado en la función *appMain_displayUserMenu* del fichero *appMain.c*.



```
Hello, Ricard!
-----
Please, select a menu option:
1. View my trips
2. List all trips
3. Publish new trip
0. Logout
```

Tarea 1

Modificad la función `appMain_displayUserMenu` para que entre en un bucle que gestionará la interacción con el usuario.

Para mostrar la información al usuario y capturar la opción seleccionada, podéis llamar a la función `appMenu_userMenu` del fichero `appMenu.c`.

La operación saldrá del bucle, con lo que retornará a la función principal, cuando el usuario apriete la tecla '0'.

Implementad la funcionalidad del bucle teniendo en cuenta las 3 opciones disponibles:

1. En caso que el usuario seleccione '1', se llamará a la función `appMenu_viewMyTrips`,
2. En caso que el usuario seleccione '2', se llamará a la función `appMenu_listAllTrips`.
3. En caso que el usuario seleccione '3', se llamará secuencialmente primero a `appMenu_publishNewTrip` y posteriormente a `appMenu_displayTrip`.

Nota: Las 2 primeras funciones (`appMenu_viewMyTrips` y `appMenu_listAllTrips`) serán implementadas en prácticas posteriores.

Tarea 2

Modificad el bucle principal de la aplicación implementada en `appMain_displayMainMenu` para que, si el usuario autenticado es correcto, llame a la función `appMain_displayUserMenu` previamente implementada.