



## 08GIIN Metodología de Programación

### Proyecto HardVIU

El proyecto de este semestre estará basado en la simulación simplificada de una transnacional encargada de ensamblar equipos informáticos bajo demanda.

Estamos en la segunda asignatura de programación del grado, simplificaremos varias cosas para no aumentar innecesariamente la dificultad. Entre ellas, toda la interacción con el programa se realizará por línea de comandos.

Menú, al iniciar al programa se mostrará al usuario las diferentes opciones en modo texto y se le permitirá elegir cualquiera de ellas.

El menú permitirá 8 opciones:

- 1) Componentes
- 2) Equipos
- 3) Distribuidores
- 4) Despachar
- 5) Días
- 6) Info sistema
- 7) Ficheros
- 0) Salir

#### 1) Componentes

##### a. Alta

Se pedirá al usuario los datos del Componente:

- Un identificador alfanumérico (nombre). Debe ser único, deben verificar que no existe ya en el sistema. (alfanumérico, mínimo 3 caracteres).
- Tipo de componente (enumerado, [Fuente, PB, TG, CPU, RAM, Disco]
- Peso en gramos (entero >0).
- Coste en euros (real > 0).
- Cantidad (entero > 0).

Una vez introducida la información, preguntará si queremos introducir otro o salir al menú principal.

Datos de ejemplo:

"i9-13900KS", CPU, 100, 649,99

"i5-13600HX ", CPU, 98, 299,99



### **b. Modificación**

Se pedirá al usuario el identificador o se le dará la opción de listar todos los componentes que hay en sistema (mostrando los identificadores).

En el caso de pedir listado, después de este se pedirá al usuario un identificador.

Una vez seleccionado el componente, se pedirá que quiere hacer con él: Cambio Stock (modificar la cantidad), Cambio información (modificar todos los datos menos el identificador), dar de baja (quitar del sistema por ser un componente discontinuado).

## **2) Equipos**

### **a. Alta**

Se pedirá al usuario los datos del equipo:

-Identificador alfanumérico único. (alfanumérico, mínimo 3 caracteres)

A continuación para cada tipo de componente dado de alta en el sistema y que exista stock, se mostraran al usuario para que seleccione uno de cada tipo. De no existir un mínimo de uno de cada tipo o si el usuario decide que no quiere continuar la configuración por no disponer de los componentes que quiere, debe poder salir al menú principal.

Para simplificar no verificaremos compatibilidades entre componentes.

Una vez introducido el equipo correctamente, preguntará si queremos introducir otro o salir al menú principal. Se debe actualizar el stock.

### **b. Modificación**

Se pedirá al usuario el identificador o se le dará la opción de listar todos los equipos que hay en sistema (mostrando los identificadores).

En el caso de pedir listado, después de este se pedirá al usuario un identificador.

Una vez seleccionado el equipo, se pedirá que quiere hacer con él: Cambio configuración (alta pero con datos actuales, se debe modificar stock de los componentes cambiados, desensamblar (quitar del sistema por ser un equipo discontinuado, recuperar stock de componentes).

## **2) Distribuidores**

### **a. Alta**

Se pedirá al usuario los datos del distribuidor:

-Nombre (alfanumérico único)



- Tiempo de entrega desde fábrica en días (entero > 0)
- Dirección (campo de texto, máximo 100 caracteres)

Una vez introducido, preguntará si queremos introducir otro o salir al menú principal.

Datos ejemplo:

"Barcelona", 10, "Consell de Cent 123, Local 4, 08015, Barcelona, Barcelona"

#### **b. Modificación**

Se pedirá al usuario el identificador o se le dará la opción de listar todos los distribuidores que hay en sistema (mostrando los identificadores).

En el caso de pedir listado, después de este se pedirá al usuario un identificador.

Una vez seleccionado el distribuidor, se pedirá que quiere hacer con él: Cambio información (modificar todos los datos menos el identificador), dar de baja (quitar del sistema, de tener equipos en despacho deben regresarse a fábrica).

#### **4) Despachar**

Se pedirá al usuario el distribuidor los datos del distribuidor y el equipo:

- Id distribuidor (debe existir).
- Id Equipo (debe existir)
- Se debe guardar que se ha hecho la asignación: Quitar de equipos disponibles, inicializar el contador de días para despacho según el tiempo de entrega del distribuidor.

#### **5) Días**

Pide al usuario la cantidad de días que van a transcurrir (entero > 0), mínimo 1 día.

Por cada día indicado debe restarse 1 en el tiempo de despacho. De llegar a 0 debe indicar al usuario que el distribuidor ha recibido el equipo.

Debe guardarse el histórico de equipos entregados para Info Sistema.

#### **6) Info Sistema**

Muestra la información del estado completo del sistema actual y el histórico:

- Componentes con toda su información, equipos con toda su información (stock, despachando y histórico de los despachados), Distribuidores con toda su información.



## 7) Ficheros

### Cargar datos

El programa debe pedir el nombre del fichero (lo buscará por defecto en la raíz donde se está ejecutando), del cual leerá los datos contenidos en el mismo.

En la documentación deben indicar el método que usaran para separar los datos: por ejemplo todos seguidos o separados por líneas, en cada línea separados por ' ',';'...' ... el separador que ustedes elijan.

### Guardar datos

El programa debe guardar el estado actual de la ejecución en un fichero (preguntar el usuario el nombre del mismo). Debe seguir el mismo formato que se utiliza para cargar datos.

## 0) Salir

Sale del programa.

Deben verificarse que todos los datos introducidos por el usuario no provoquen un error (por ejemplo introducir "hola" cuando se espera un entero). Tener especial cuidado en controlar los posibles errores producidos al trabajar con ficheros. Se debe usar estructuras Try - Catch donde sea necesario.

Todas las opciones deben incluir vía de escape, para evitar que el programa se quede bloqueado. Por ejemplo, si al crear un equipo hay un tipo de componente del cual no hay en stock se debe dar un mensaje de error y regresar al menú principal, no se puede quedar en bucle pidiendo un componente.

La entrega final debe incluir modularidad.

Antes de empezar a programar es muy recomendable hacer un esquema de cómo pensáis solucionar el problema y que estructuras de datos utilizareis.

Hay varias maneras de realizar el proyecto, cualquiera que cumpla con lo que se pide a nivel funcional y que siga las pautas de documentación y estructura es válida.

El proyecto es individual, los códigos serán verificados que no tengan similitudes excesivas entre estudiantes.

Un punto donde podéis elegir es en el tipo de estructuras de datos (tablas, listas, diccionarios, ...) que necesitareis para guardar todos los parámetros del sistema que se piden, en principio cualquier aproximación que se haga es correcta, no tendréis más o menos puntos por usar una u otra. Elegid la que veáis más fácil.

Se requiere documentar el código y será parte de la nota.



### **Ampliación (puntos adicionales, máximo 1):**

Algunos de vosotros tenéis un conocimiento mayor del Python y de programación. Podéis complicar un poco el proyecto añadiendo alguna funcionalidad adicional. Lo consideraré como trabajo extra y será recompensado con una nota superior al 10. Los cambios no deben modificar la funcionalidad básica (la descrita en esta actividad), o en ese caso la ampliación puede hacer que la nota sea menor a 10 por no cumplir lo que se pide. Ejemplo de posible ampliación, hacer los menús de interacción de usuario especialmente trabajados usando colores y símbolos decorativos.

## **Entrega**

Se debe entregar los el código en un .zip (librerías utilizadas + código generado en python dividido en módulos)y un PDF explicativo con un esquema de cómo se pretende solucionar el problema y un breve explicación de las estructuras de datos utilizadas.

**Deben incluirse en el PDF el formato utilizado para guardar y recuperar los datos.**

Tanto el PDF como cada uno de los ficheros de código, deben estar claramente identificados con el nombre de la asignatura y del estudiante.

### **Entregas parciales**

Hay hasta dos entregas parciales antes de la final. Les recomiendo utilizarlas para tener feedback. En ellas puede no estar implementado todo.

Por ejemplo pueden entregar:

Primera entrega con el esquema/documentación y la lógica interna funcionando (menú del 1 al 4 y el 0).

En una segunda entrega añadir los puntos 5 y 6 el menú.

Entrega final con todo.