**Relación de entidades del proyecto**

Despacho

Componente

Equipo

Distribuidor

De colecciones tenemos:

* Lista de componentes de almacén (lista de Componente)
* Lista de equipos disponibles (lista de Equipo)
* Lista de distribuidores (lista de Distribuidor)
* Diccionario de equipos despachando. Relaciona como claves el identificador de un distribuidor, y como valores los despachos de dicho distribuidor. Cada despacho contiene el equipo que se está despachando y el tiempo restante en días.
* Lista de equipos despachados (lista de Equipo)

**Explicación de módulos**

* Main.py
  + Módulo principal donde se ejecuta el menú principal. En función de la opción introducida por el usuario, se llamará a las funciones presentes en cada uno de los módulos, divididos por funciones
* Utiles.py
  + Módulo con funciones de utilidad, que se usan en distintos puntos del proyecto
  + Para imprimir por pantalla mensajes de colores (ampliación del proyecto) se ubican en este módulo funciones con nombre *print…*, de tal manera que es en este módulo donde se importa la librería colorama. Así se consigue que el resto de módulos se abstraigan de dicha librería, y tan sólo se acoplan con el módulo *utiles.py*
* Estructuras.py
  + Módulo que contiene las colecciones de datos (componentes, equipos despachados, etc), además de las definiciones de las clases de cada entidad. La lista de entidades es como sigue:
    - Componente
    - Equipo
    - Distribuidor
    - Despacho
* Testing.py
  + Módulo con datos de prueba, para realizar tests de la aplicación a lo largo del desarrollo realizado. Se podría haber eliminado de cara a la entrega, pero lo dejamos para tener distintos datos de entrada a la hora de ejecutar la aplicación
* Componentes.py
  + Módulo referente a la funcionalidad de componentes, opción 1 del menú principal. Tanto este módulo como el de equipos y distribuidores siguen un esquema similar, ya que comparten como patrón el alta y modificación de sus datos.
  + Cada uno de estos módulos tiene definidos controles de errores para la introducción de datos, tanto para la introducción del identificador como de la información restante (stock, tiempo restante, etc)
  + Para un componente, no se podrá modificar ni su identificador ni su tipo una vez creado en el sistema. Sólo serán modificables los datos de peso, coste y cantidad.
* Equipos.py
  + Módulo que implementa la funcionalidad del alta y modificación de los equipos, opción 2 del menú principal
  + Un equipo dispone de un identificador único (comprobación idéntica que para el identificador de un componente) y de una lista de componentes.
  + Se decide implementar la lista de componentes de un equipo en formato diccionario, para así agrupar el tipo del componente (clave del diccionario) con su valor (objeto componente, referenciando al componente de la lista de componentes disponibles)
  + El equipo dispondrá en su listado de componentes de las referencias a los objetos originales. Esto quiere decir que si, una vez ensamblado un equipo, si uno de sus componentes es modificado en la opción 1 del menú principal, el cambio se verá reflejado en dicho equipo.
  + A la hora de crear un equipo se comprobará si existen componentes para cada tipo, al menos uno, y por supuesto que exista stock (función checkComponentes)
* Distribuidores.py
  + Módulo que implementa la funcionalidad del alta y modificación de un distribuidor.
  + La información del distribuidor es parecida a la del componente; identificador único y una serie de datos descriptivos
* Despachar.py
  + Módulo que incluye la funcionalidad de la acción de despachar (opción 5) y la de Días (opción 6)
  + La función despachar creará un nuevo despacho en el diccionario de Despachos, relacionando el identificador del distribuidor seleccionado con un nuevo objeto Despacho, en el cual se tiene, además del objeto distribuidor referenciado, una referencia al equipo que se va a despachar y un campo numérico para indicar los días restantes que faltan
  + La función *días* recorre la estructura completa de Despachos y resta el tiempo restante a los días especificado por el usuario. Para cada despacho cuyo tiempo restante sea 0 o menor, se eliminará de la estructura de despachos, y el equipo se incluirá en la lista de equipos despachados
* Info.py
  + Módulo que incluye las funciones de mostrar la información del sistema (opción 6) y de ficheros (opción 7)
  + La función *info\_sistema* vuelca en pantalla toda la información referente a las estructuras principales, que son:
    - Lista de componentes de almacén
    - Lista de equipos disponibles
    - Lista de distribuidores
    - Lista de equipos despachados
    - Información de despachos
  + Para mostrar toda esta información se definieron funciones *\_\_str\_\_*, para representar como cadenas cada una de las instancias a las clases Componente, Equipo, Distribuidor y Despacho
  + La función *ficheros* nos permite elegir entre la opción *cargar* o *guardar* en un fichero de texto. El nombre del fichero se pude al usuario por teclado, controlando que no se introduce un nombre vacío, como en el resto del proyecto. Además, existe un try/catch para controlar si fichero se abre correctamente, tanto en la carga como en el guardado. Sobre todo en la carga, para ver si el fichero existe.
  + El formato del fichero es el que sigue:

Listado de cada componente, con sus datos separados por el carácter ‘;’

---

Listado de equipos disponibles, también separados por ‘;’

---

Listado de equipos despachados

---

Listado de distribuidores

---

Listado de despachos, esto es, equipos que se están despachando actualmente

---

Un ejemplo de fichero sería el siguiente:

c4;CPU;50;10.5;7

c5;RAM;50;10.5;7

c6;Disco;50;10.5;8

c10;Fuente;10;10.5;10

c20;PB;10;10.5;10

c30;TG;10;10.5;10

c40;CPU;10;10.5;10

c50;RAM;10;10.5;10

c60;Disco;10;10.5;9

---

e1;Fuente-c1;PB-c2;TG-c3;CPU-c4;RAM-c5;Disco-c6

---

e2;Fuente-c1;PB-c2;TG-c3;CPU-c4;RAM-c5;Disco-c6

---

dist1;Barcelona;10;Consell de Cent 123, Local 4, 08015, Barcelona, Barcelona

dist2;Sevilla;4;Avenida de José Laguillo, Sevilla, Sevilla

---

dist1;e3;Fuente-c1;PB-c2;TG-c3;CPU-c4;RAM-c5;Disco-c6;5

---

**AMPLIACIÓN**

* Se incluye funcionalidad en el módulo *utiles.py* para mostrar diferentes colores en función del tipo de mensaje mostrado (funciones print…). En general, se usa el color rojo para indicar errores al usuario, blanco como Log y azul para mostrar elementos del menú
* Otra funcionalidad añadida es darle al usuario la facilidad a la hora de modificar un componente, equipo o distribuidor; por cada atributo a modificar, se mostrará por pantalla el valor anterior del campo y se le pedirá el nuevo. Si el usuario pulsa INTRO, se mantendrá el valor anterior en dicho campo.