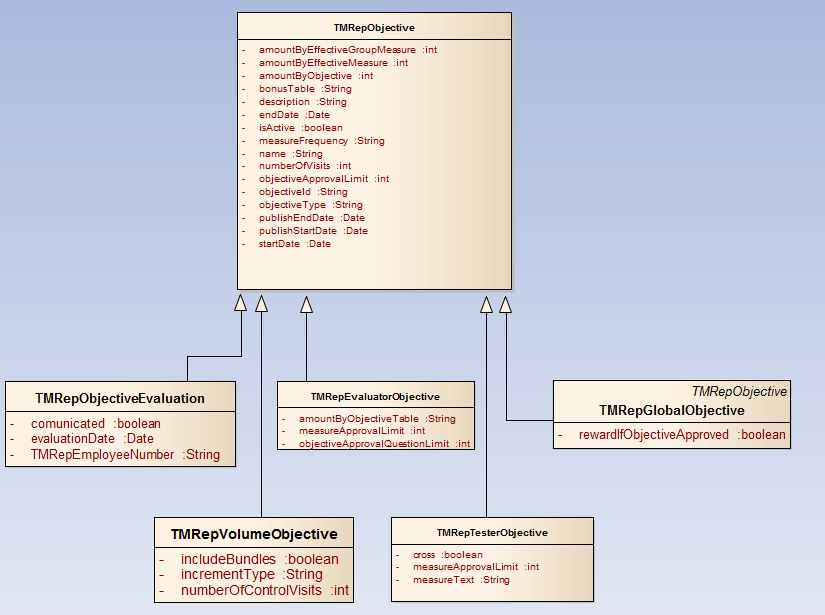
1. Crear una una clase llamada **Password** que siga las siguientes condiciones:
   * Que tenga los atributos **contraseña el cual sera un array de chars**. La longitud será de 8.
   * Los constructores serán los siguiente:
   * Un constructor por defecto.
   * Un constructor que reciba un String, solamente se tomaran los primeros 8 caracteres de la cadena, de ser menos caracteres se completaran los espacios vacíos con @
   * Los métodos que implementa serán:
     1. **esFuerte()**: devuelve un booleano si es fuerte o no, para que sea fuerte debe tener de 2 mayúsculas, 1 minúscula y 5 números.
     2. **generarPassword()**:  genera la contraseña aleatoria usando letras y nros, se tendrán tres arrays, uno por letras minúsculas, otro mayusculas y otro con nros. Para cada posición se usara la opción random para elegir un arreglo y luego con la misma función se elegirá una posición. Hasta formar la password
     3. Método **get** para contraseña, deberá devolver un String resultante de concatenar todos los caracteres de la password
     4. Método **set**: Se comportara igual que el constructor.
   * crear una clase ejecutable que haga lo siguiente:
     1. Crea un array de Passwords de valor fijo
     2. Crea un metodo permita al usuario elegir entre crear la password automáticamente o que lea una password desde la pantalla y cree un objeto para cada posición del array.
     3. Crea otro array de booleanos donde se almacene si el password del array de Password es o no fuerte (usa el bucle anterior).
     4. Al final, muestra la contraseña y si es o no fuerte (usa el bucle anterior). Usa este simple formato:
     5. contraseña1 valor\_booleano1
     6. contraseña2 valor\_bololeano2
2. crear una clase llamada **Serie** con las siguientes características:
   * Sus atributos son **titulo, numero de temporadas**, **entregado, genero y creador.**
   * Por defecto, el número de temporadas es de 3 temporadas y entregado **false**. El resto de atributos serán valores por defecto según el tipo del atributo.
   * Los constructores que se implementaran serán:
     1. Un constructor por defecto.
     2. Un constructor con el titulo y creador. El resto por defecto.
     3. Un constructor con todos los atributos, excepto de entregado.
   * Los métodos que se implementara serán:
     1. Métodos get de todos los atributos, excepto de entregado.
     2. Métodos set de todos los atributos, excepto de entregado.
     3. Sobrescribe los métodos toString.
   * crear una clase **Videojuego** con las siguientes características:
     1. Sus atributos son **titulo, horas estimadas, entregado, genero y compañia**.
     2. Por defecto, las horas estimadas serán de 10 horas y entregado false. El resto de atributos serán valores por defecto según el tipo del atributo.
   * Los constructores que se implementaran serán:
     1. Un constructor por defecto.
     2. Un constructor con el título y horas estimadas. El resto por defecto.
     3. Un constructor con todos los atributos, excepto de entregado.
   * Los métodos que se implementara serán:
     1. Métodos get de todos los atributos, excepto de entregado.
     2. Métodos set de todos los atributos, excepto de entregado.
     3. Sobrescribe los métodos toString.
   * Como vemos, en principio, las clases anteriores no son padre-hija, pero si tienen en común, por eso vamos a hacer una interfaz llamada **Entregable** con los siguientes métodos:
     1. **entregar()**: cambia el atributo prestado a true.
     2. **devolver()**: cambia el atributo prestado a false.
     3. **isEntregado()**: devuelve el estado del atributo prestado.
   * Crear una aplicación ejecutable y realizar lo siguiente:
     1. Crear dos arrays, uno de **Series** y otro de **Videojuegos**, de 5 posiciones cada uno.
     2. Crea un objeto en cada posición del array, con los valores que desees, puedes usar distintos constructores.
     3. Entrega algunos **Videojuegos** y **Series** con el método **entregar()**.
     4. Cuenta cuantos **Series** y **Videojuegos** hay entregados. Al contarlos, devuélvelos.
     5. Por último, indica el **Videojuego** tiene más horas estimadas y la serie con más temporadas. mostrar en pantalla con toda su información (usa el método toString()).
3. Crear un array de nros de 10 posiciones, leer todas las posiciones de por teclado, una vez finalizados se deberá invertir el orden de los mismos
4. Definir las siguientes clases:
   * Clases:
     1. Serie: tendrá un director, un booleano si es cómica o no
     2. Noticiero: Tendrá un tema (tecnología, Actualidad, Moda), un conductor
     3. Largometraje: título, director
     4. Documental: temática y director
     5. Recital: artista, director
     6. Visualizable: Una interfaz que tenga el método visualizar
     7. Televisor: Tiene marca, tamaño de pantalla
   * Métodos:
     1. Para todas las clases salvo televisor crear el método visualizar: mostrar en pantalla los datos de la clase ejemplo “estoy reproduciendo el largometraje X del director X
     2. Todas las clases salvo televisor deben implementar la interfaz visualizable
     3. Un constructor que inicialice los datos de las clases
     4. La clase televisor deberá tener un método que reciba un objeto que implemente visualizable y lo muestre visualice en pantalla
   * Crear un programa que haga lo siguiente
     1. Crear un array de medios de 6 posiciones
        + Solicitar al usuario que cargue cada una de las 6 posiciones siendo
          1. S o s: Serie
          2. N o N: Noticiero
          3. L o l: Largometraje
          4. D o d: documental
          5. R o r: recital
     2. Recorrer el array y visualizar cada uno de los medios introducidos en el punto anterior
5. Implementar el siguiente modelo 
6. Crear en el Enterprise Architect las clases del ejercicio 4 y sus relaciones