

Índice

1. Descripción del problema	1
1.1. Clase Usuario	1
1.2. Clase Menu	3
1.3. Clase SistemaGestionUsuarios	4
2. Tareas	6

1. Descripción del problema

En la construcción de un Portal Web, una de las tareas a realizar es la definición del subsistema de gestión de usuarios. La gestión de usuarios que vamos a tratar contempla mostrar un menú de opciones personalizado al usuario, una vez éste ha llevado acabo la autenticación de manera satisfactoria.

La interacción del usuario con el sistema presenta los siguientes pasos:

- El usuario se autentifica ante el sistema introduciendo login y password.
- El sistema de gestión de usuarios ante un login y password correcto muestra al usuario el menú de operaciones que puede realizar, y un mensaje de autenticación incorrecta en caso contrario.

El modelado del problema se puede realizar con las siguientes clases:

- Usuario.
- Menu.
- SistemaGestionUsuarios.

1.1. Clase Usuario

La clase Usuario recoge la información referente a un usuario potencial del Portal Web: login, su password y su nombre.

```
/*
 * Created on 25-02-2016
 *
 * (c) Departamento de Informática
 * Universidad Europea del Atlántico
 * 2016
 */
/**
 * Usuario del Portal Web.
 *
 * @author <A href="mailto:juan.tortajada@uneatlantico.es">Juan Tortajada</A>
 *
 * @version 1.0
 */
public class Usuario {

    /**
     * Nombre del usuario.
     */
    private String nombre;

    /**
     * Login/identificador del usuario.
     */
    private String login;

    /**
     * Contraseña del usuario.
     */
    private String password;

    /**
     * Crea un usuario a partir de su nombre, login y password.
     * @param n Nombre del usuario.
     * @param l Login del usuario.
     * @param p Contraseña del usuario.
     */
    public Usuario (String n, String l, String p) {...}

    /**
     * Obtiene el login del usuario.
     *
     * @return Login.
     */
    public String getLogin() {...}

    /**
     * Obtiene el nombre del usuario.
     * @return Nombre.
     */
    public String getNombre() {...}

    /**
     * Obtiene la contraseña.
     * @return Contraseña.
     */
    public String getPassword() {...}
}
```

1.2. Clase Menu

La clase menú contiene una lista finita de opciones, y métodos para añadir y obtener los valores de las mismas.

```
/*
 * Created on 25-02-2016
 *
 * (c) Departamento de Informática
 * Universidad Europea del Atlántico
 * 2016
 */

/**
 * Menu de operaciones.
 *
 * @author <A href="mailto:juan.tortajada@uneatlantico.es">Juan Tortajada</A>
 *
 * @version 1.0
 */
public class Menu {

    /**
     * Opciones del menú.
     */
    String opciones[];

    /**
     * Número total de opciones.
     */
    int nOpciones;

    /**
     * Primera opción libre.
     */
    int opcionLibre;

    /**
     * Crea una menú de opciones con un número de opciones.
     * @param nO Número de opciones.
     */
    public Menu(int nO) {...}

    /**
     * Añade una opción al menú.
     * @param op Opción del menú.
     */
    public void addOpcion(String op) {...}

    /**
     * Obtiene sólo las opciones introducidas en el menú.
     * @return Opciones del menú.
     */
    public String[] getOpciones() {...}
}
```

1.3. Clase SistemaGestionUsuarios

```
/*
 * Created on 25-02-2016
 *
 * (c) Departamento de Informática
 * Universidad Europea del Atlántico
 * 2016
 */

/**
 * Menu de operaciones.
 *
 * @author <A href="mailto:juan.tortajada@uneatlantico.es">Juan
Tortajada</A>
 *
 * @version 1.0
 */
public class SistemaGestionUsuarios {

    /**
     * Número máximo de usuarios.
     */
    private int nUsuarios;

    /**
     * Primera posición libre.
     */
    private int primerLibre;

    /**
     * Usuarios del sistema.
     */
    private Usuario[] usuarios;

    /**
     * Menús asociados a los usuarios.
     * El usuarios[i] tiene asociado el menus[i].
     */
    private Menu[] menus;

    /**
     * Crea un sistema de gestión de usuarios limitado en número.
     * @param nU Número total de usuarios.
     */
    public SistemaGestionUsuarios(int nU) {...}

    /**
     * Añade un usuario al sistema y le asocia un menu de
opciones.
     * @param u Usuario.
     * @param m Menú de opciones.
     */
    public void addUsuario(Usuario u, Menu m) {...}

    /**
```

```
* Autentifica un usuario, y devuelve su menú si todo va bien,  
en caso  
* contrario devuelve <i>null</i>.  
*  
* Recorre la lista de usuarios, buscando un usuario con el  
mismo login que  
* el suministrado, si lo encuentra comprueba que los  
passwords coinciden,  
* devolviendo el menú que se encuentra en la misma posición.  
*  
* @param login Login de la autenticación.  
* @param password Contraseña de la autenticación.  
* @return Menú asociado al usuario o <i>null</i>.  
*/  
public Menu autentificaUsuario(String login, String password)  
{  
    ...  
}
```

2. Tareas

Se piden las siguientes tareas:

1. Crear un proyecto en Eclipse con el código suministrado (clases y librerías).
2. Implementar las clases que se suministran incompletas.
3. Implementar una clase de test `LoginTest` que se encargue de crear un sistema de gestión de usuarios, registre una serie de usuarios con menú's personalizados, y pruebe una autenticación exitosa (listando las opciones del menú) y otra fallida (mostrando un mensaje por consola).
4. Ejecutar la aplicación.
5. Usar el depurador. Colocar un *breakpoint* en una línea de código para observar el valor de las variables locales.
6. Añadir comentarios de documentación apropiados a todas las clases y todos los métodos. Generar la documentación desde Eclipse.
7. Exportar en un fichero zip el trabajo realizado. Borrar el proyecto desde Eclipse, junto con los contenidos.