

IMPORTANTE:

- **AQUELLOS PROYECTOS QUE PRESENTEN ERRORES DE COMPILACIÓN O DE EJECUCIÓN NO SERÁN CORREGIDOS**
- *El proyecto debe utilizar el patrón Modelo-Vista-Controlador.*
- *Pon a los métodos los nombres que se indican.*
- *Todas las rutas deben ser relativas.*

INSTRUCCIONES DE ENVÍO:

1. Crea una carpeta con el nombre “examenR1_NOMBRE_APELLIDO”.
2. En la carpeta anterior, crea un proyecto con el nombre “examenR1_NOMBRE_APELLIDO”.
3. Cuando indique el profesor, cierra NetBeans.
4. Comprime en formato **ZIP** la carpeta del proyecto y súbelo a la entrada indicada en la Plataforma.

TEMÁTICA

Nos ha contratado el ayuntamiento de una localidad para gestionar los datos de la realización de una oposición.

opositores.txt

12345678A;García Retamero;Javier
23456789B;López Martínez;Ana
34567890C;Fernández Gómez;Carlos
45678901D;Ruiz Sánchez;Lucía
56789012E;Torres Delgado;Miguel
67890123F;Ramírez Ortega;Sofía
78901234G;Moreno Castro;David
89012345H;Jiménez Navarro;Laura
90123456I;Pérez Molina;Andrés
01234567J;Gómez Herrera;Elena
11223344K;Vargas León;Manuel
22334455L;Domínguez Ríos;Clara
33445566M;Santos Aguado;Pablo
44556677N;Cano Nieto;Isabel
55667788O;Ortega Salas;Raúl
66778899P;Delgado Muñoz;Teresa
77889900Q;Castro Medina;Jorge
88990011R;Navarro Rubio;Beatriz
99001122S;León Cabrera;Hugo
10111213T;Molina Vázquez;Natalia

respuestas.txt

12345678A;abdcdbacda
23456789B;dcabcbadcd
34567890C;bacdababc
45678901D;cadbacdbbd
56789012E;
67890123F;acdbbdacdd
78901234G;bdacdadbba
89012345H;cbadbdcac
90123456I;
01234567J;adcbadcbdb
11223344K;
22334455L;ddbcbacda
33445566M;cdbadcbadb
44556677N;abdcadcbdd
55667788O;
66778899P;bacdbacdad
77889900Q;cadcadbcab
88990011R;
99001122S;acdbacdbca
10111213T;bdacbdacbd

Plantilla_correctora.txt

bdacdbacdc

EJERCICIO 0: Crea una Vista con el siguiente menú:

La aplicación tendrá una interfaz en modo texto y debe tener las siguientes opciones:

=====

0. Salir

1. FASE 1 (Corregir exámenes)

2. FASE 2 (Modificar respuestas)

3. FASE 3 (Generar archivo resultados)

4. FASE 4 (Copias de seguridad)

=====

Introduzca la opción:

No se saldrá del menú hasta que se pulse la opción 0.

Cada vez que se ejecute una opción, se volverá a este menú.

FASE 1: Corrección de exámenes (3,5 puntos):

Nombre método Modelo: **correccionExamen**

Parámetros de entrada: **Archivo de respuestas, plantilla correctora**

- Una vez finalizada la prueba de oposición, el sistema deberá realizar la corrección de los exámenes. Para ello, nos han pasado un fichero (**respuestas.txt**) con las respuestas de cada opositor (10 letras, 1 por cada pregunta), un archivo con la plantilla correctora (**plantilla_correctora.txt**) y un archivo con los datos personales (**opositores.txt**). En el archivo de respuestas, si un opositor no se ha presentado al examen aparece su DNI pero no sus respuestas.
- Tendremos que generar un **archivo binario de acceso aleatorio** (no con objetos) denominado **datos_finales.dat**, en la carpeta **binario_acceso_aleatorio**, con los datos de cada opositor. De cada opositor guardaremos su **dni, apellidos, nombre, las respuestas que puso en el examen y la nota obtenida** (número de letras que coinciden entre la respuesta del opositor y la plantilla). Ten en cuenta que hay opositores que no se han presentado.
- El **método** creará, partiendo del fichero binario, un fichero de texto (**datos_finales.txt en la carpeta binario_acceso_aleatorio**) con el contenido de todo el archivo binario generado.

FASE 2: Reclamaciones (3,5 puntos):

Nombre método Modelo: **modificarRespuestas**

Parámetros de entrada: **Id del opositor, cadena con 10 letras de la "a" a la "d" correspondiente a las respuestas.**

- Los opositores tienen un período de reclamación si han detectado que el sistema ha registrado una respuesta de forma errónea. Es por ello que tenemos que poder modificar las respuestas en el fichero binario de acceso aleatorio para un determinado opositor. Le pasaremos el **id del opositor** del archivo binario y las respuestas. El sistema recalculará la nota de forma automática comparándola con la plantilla correctora y actualizará tanto las respuestas como la nota en el archivo binario.
- Si no existe un opositor con ese ID, mostrará un mensaje indicándolo.
- El método devolverá los datos del opositor con ese ID, tras hacer una lectura del archivo binario.
- La vista debe mostrar por pantalla los datos devueltos.

FASE 3: Publicación de resultados finales (2 puntos):

Nombre método Modelo: **publicaResultados**

- El cliente quiere que los resultados se publiquen en una página Web. Para ello, el sistema deberá generar en memoria un árbol DOM-XML del archivo binario generado en apartados anteriores.
- Modifica el árbol DOM para que, sólo aparezcan los que tengan una nota ≥ 5 .
- Genera el archivo XML (**resultados_finales.xml en la carpeta xml_html**) a partir del árbol DOM.
- Genera un archivo HTML (**resultados_finales.html en la carpeta xml_html**) para la publicación.

FASE 5: Copias de seguridad (1 punto):

Nombre método Modelo: **copiaSeguridad**

- Finalmente, se hará una copia de seguridad de todos los archivos, tanto los que nos habían suministrado como los que hemos generado nosotros, en una carpeta (**copia_seguridad**) que tendrá la misma estructura y los mismos nombres que la carpeta de partida.