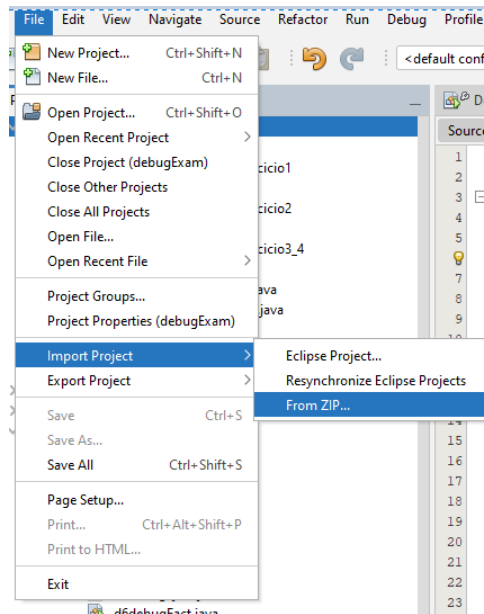


DAM – Entornos de desarrollo		
Examen		CURSO : 2022-23

Entornos de desarrollo

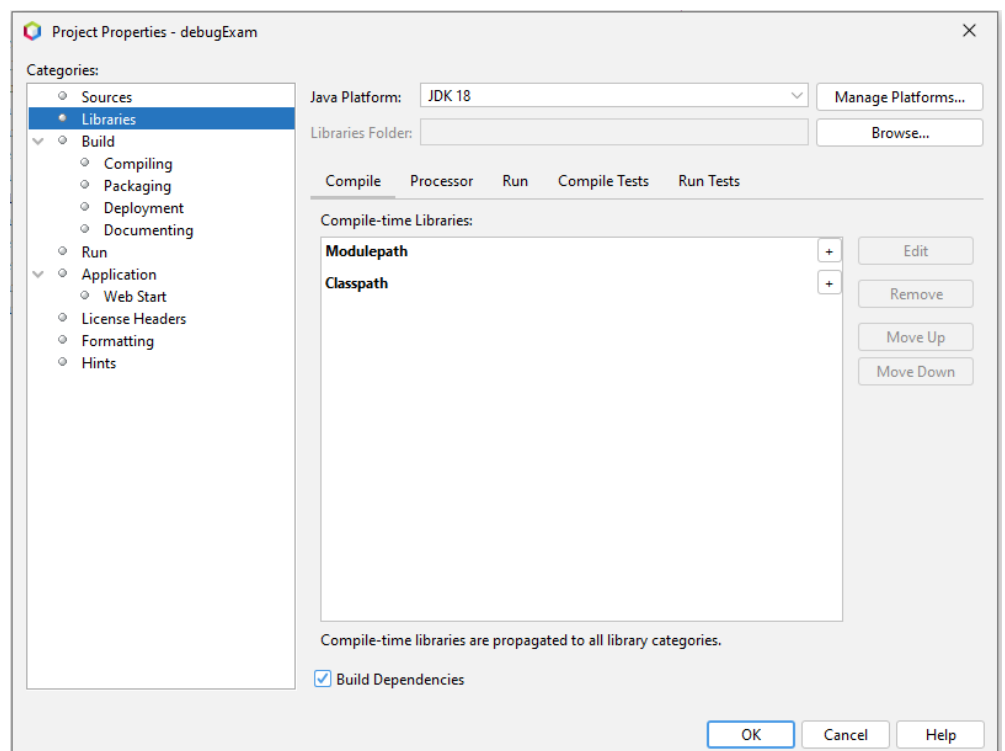
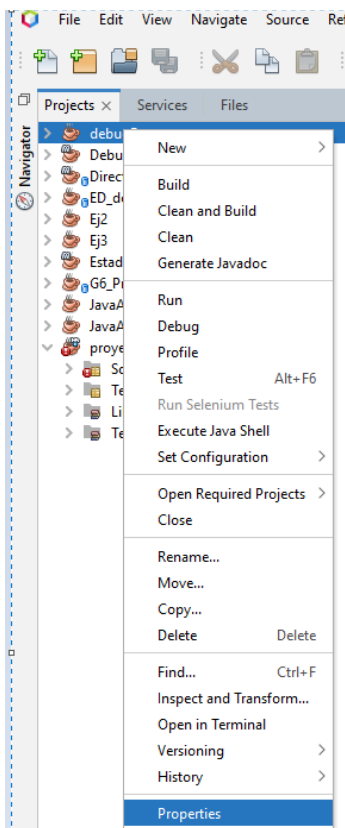
(4 pts) EJERCICIOS DEBUGGING

Descargar el fichero **debugExam.zip** que encontrarás en la plataforma aules e impórtalo a **Netbeans** mediante la opción File → Import Project → From ZIP...

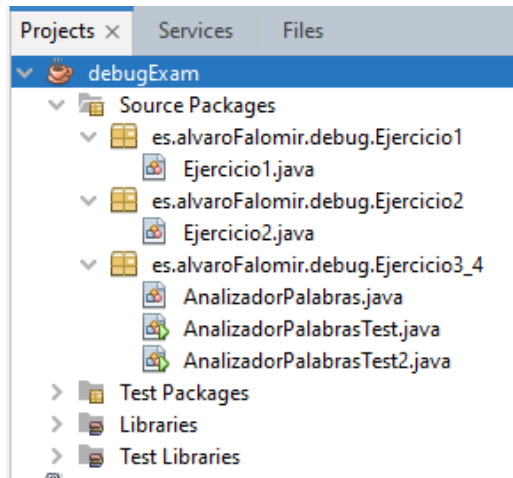


(WARNING) Si al importar el proyecto te aparece un mensaje de **warning (icono triangulo amarillo)** sobre el proyecto, debes realizar las siguientes acciones:

- Hacer click con el botón derecho del ratón sobre el proyecto y seleccionar la opción **properties**.
- En la siguiente pantalla seleccionar **Libraries** y elegir la versión de la máquina virtual JAVA instalada.



Dentro del proyecto Java que acabas de importar encontrarás 4 ejercicios para depurar:



- **(1 pts)** Ejercicio1.java
 - Debes encontrar el error y arreglar el programa para que realice la cuenta atrás correctamente tal y como se te indica en los comentarios dentro de la clase Java.

- **(1 pts)** Ejercicio2.java
 - Debes realizar:
 - Crear 1 **breakpoint** en la línea 16 y **realizar una captura de pantalla** cuando el programa este parado en esa línea en la que se muestre como se ha modificado la variable aleatorio desde el explorador de variables para que tenga el valor 100000.
 - Crear 1 **breakpoint** en la línea 19 y **realizar una captura de pantalla** cuando el programa este parado en esa línea en la que se muestre como se ha modificado la variable aleatorio desde el explorador de variables para que vuelva a tener el valor que tenía tras la ejecución de la función **devuelveNumeroAleatorio(80, 160)** de la línea 8.
 - Crear 1 **breakpoint** condicional en la línea 22 que haga que el programa se pare en esta línea si se cumple la condición de que la variable **i** tenga el valor 60 y la variable **j** tenga el valor 75.
Realiza una captura de pantalla en el que se vea como has definido el **breakpoint** condicional.
Realiza una captura de pantalla cuando el programa este parado en la línea 22 en la que se muestre el valor de la variable sum mediante el explorador de variables.

- (1 pts) Ejercicio3.java

```
1 package es.alvaroFalomir.debug.Ejercicio3_4;
2
3 public class Ejercicio3
4 {
5     public static void main(String[] args)
6     {
7         test(s:"aardvark"); // resultado esperado: a
8         test(s:"roommate"); // resultado esperado: o (no m)
9         test(s:"mate"); // resultado esperado: 0 (No hay caracteres repetidos.)
10        test(s:"test"); /* resultado esperado: 0 (No hay caracteres repetidos...
11                        * la letra t se repite pero no de manera adyacente)
12                        */
13    }
14
15    public static void test(String s)
16    {
17        AnalizadorPalabras analizador = new AnalizadorPalabras(unaPalabra: s);
18        char result = analizador.primerCaracterRepetido();
19
20        if (result == 0)
21            System.out.println(s: "No hay caracteres repetidos.");
22        else
23            System.out.println("Primer caracter repetido = " + result);
24    }
25 }
```

- En este programa se realizan varias llamadas a la función **primerCaracterRepetido()** de la clase **AnalizadorPalabras** que devuelve el primer carácter repetido de una palabra. Un carácter se considera repetido si al menos se repite 2 veces en posiciones adyacente. Por ejemplo el carácter 'l' se considera repetido en "hollow", pero el carácter 'o' no.

El programa realiza varias llamadas a esta función a través de la función **test()** obteniendo resultados incorrectos.

Debes encontrar el error y arreglar el código de la función **primerCaracterRepetido()** para que las 4 llamadas a la función **test()** devuelvan el resultado esperado.

- **(1 pts)** Ejercicio4.java

```

Source History
1 package es.alvaroFalomir.debug.Ejercicio3_4;
2
3 public class Ejercicio4
4 {
5     public static void main(String[] args)
6     {
7         test(s: "mississippiii", expected: 4); // resultado esperado: 4 (ss, ss, pp, iii)
8         test(s: "test", expected: 0); // resultado esperado: 0 (no hay letras repetidas)
9         test(s: "aabbcdaaaabb", expected: 4); // resultado esperado: 4 (aa, bb, aaaa, bb)
10    }
11
12    public static void test(String s, int expected)
13    {
14        AnalizadorPalabras analizador = new AnalizadorPalabras( unaPalabra: s);
15        int result = analizador.cuentaGruposCaracteresRepetidos();
16
17        System.out.println(result + " caracteres repetidos.");
18        System.out.println("Esperados: " + expected);
19    }
20 }

```

- En este programa se realizan varias llamadas a la función **cuentaGruposCaracteresRepetidos()** de la clase **AnalizadorPalabras** que devuelve el número de grupos de caracteres repetidos que se encuentran dentro de una palabra. Por ejemplo, "mississippi!!!" tiene cuatro grupos de caracteres repetidos, en este caso son: **ss, ss, pp** y **!!!**

El programa realiza varias llamadas a esta función a esta función a través de la función **test()** obteniendo resultados incorrectos.

Debes encontrar el error y arreglar el código de la función **cuentaGruposCaracteresRepetidos()** para que las 4 llamadas a la función **test()** devuelvan el resultado esperado.

SUBIR A AULES:

- **Fichero resultadoDebugExam.zip que contendrá en su interior el proyecto Java de Netbeans con los ejercicios resueltos.**
- **Fichero capturasDebugging.zip que contendrá en su interior las capturas de pantalla del ejercicio 2 de debugging.**

(6 pts) EJERCICIOS GIT


(A) (1 pt) Haciendo uso de la interfaz gráfica de **Netbeans** debes subir a tu cuenta de **github** el proyecto **debugExam** que has importado de forma manual en el primer ejercicio de este examen. El proyecto **debugExam** que subas a **github** debe contener al menos **2 commits** con cambios que hayas realizado en las clases JAVA del proyecto.

Debes realizar capturas de pantalla en las que se muestren:

- Inicialización del repositorio **git** desde **Netbeans**.
- Creación de los 2 commits desde **Netbeans**.
- Subida del proyecto al repositorio remoto en **github** desde **Netbeans**.
- Repositorio **github** con el proyecto subido.

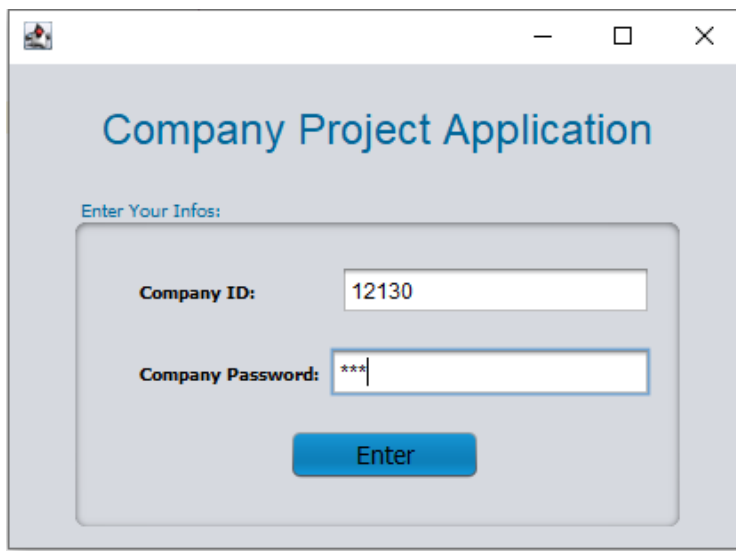
(B) (1 pt) Haciendo uso de la interfaz gráfica de **Netbeans** debes clonar el proyecto JAVA que se encuentra en la siguiente URL: <https://github.com/Vinexy/Project-Management-System-With-GUI-Using-Netbeans-JAVA.git>

(WARNING) Si al importar el proyecto te aparece un mensaje de **warning (icono triangulo amarillo)** sobre el proyecto, debes realizar las acciones indicadas en la hoja 1 de este documento.

Al ejecutar el programa mediante el botón de la flecha verde  te debe aparecer una interfaz gráfica que solicita la introducción un identificador de compañía y una contraseña.

Debes introducir los siguientes datos de acceso:

- Company ID: **12130**
- Company Password: **574**



Si has introducido correctamente los datos te debe aparecer una nueva venta para la introducción de datos bajo el título **Employee Infos**, en caso contrario la aplicación mostrará un mensaje de error indicando que la introducción de las credenciales de acceso no es correcta.

Employee Info

Name :

Surname :

ID :

Age:

Department:

Salary :

Project:

Job:

Project Process

Search or Delete Project :

Add New Employ...

Create a New Project :

Company Infos :

Acceso OK

Company Project Application

Enter Your Infos:

ID number or password is incorrect. Enter Again!!!

Company ID:

Company Password:

Credenciales de acceso incorrectas

A continuación vamos a modificar la contraseña de acceso del usuario cuyo company ID es **12130**, para ello debemos acceder al fichero **FEmployee.txt** que se encuentra dentro del proyecto que acabas de clonar desde **github**. Debes modificar el valor **574** de la línea **14** por el valor **messiCampeon**.

Source	History								
1	12000	Merve	f	48	bilisim	8856	1000	0	
2	12001	Aslı	g	32	üretim	8951	1001	1	
3	12002	Ufuk	h	28	üretim	7959	1003	2	
4	12003	Kadir	k	53	bilisim	5213	1002	3	
5	13001	Murat	i	28	test	6753	1002	4	
6	13002	Beyza	l	36	bilisim	7123	1002	5	
7	13003	Efe	m	32	test	5891	1000	6	
8	13004	Kübra	n	29	üretim	6546	1000	7	
9	13005	Fatih	t	30	test	7005	1001	8	
10	13006	Kerim	s	41	üretim	7564	1001	9	
11	13007	Sıla	y	35	bilisim	6943	1003	10	
12	13008	Dilan	z	45	üretim	5963	1003	11	
13	13009	Emre	x	35	üretim	7591	-	12	
14	12130	Dilara	Patır	28	bilisim	12000	1004	574	
15	12964	ceyda	patır	23	bilisim	5000.0	-	18	
16	13828	zeynep	doğan	34	Bilisim	3000.0	-	904	
17	13365	Ahmet	Uyan	54	Tasarım	4500.0	-	907	
18	13255	Ahmet	Yılmaz	54	Üretim	4500.0	1004	956	
19	13145	Canan	Öztürk	44	Üretim	4500.0	1004	954	
20	13348	Mesut	Yılmaz	45	Bilisim	3000.0	-	683	

FEmployee.txt antes del cambio

Source	History								
1	12000	Merve	f	48	bilisim	8856	1000	0	
2	12001	Aslı	g	32	üretim	8951	1001	1	
3	12002	Ufuk	h	28	üretim	7959	1003	2	
4	12003	Kadir	k	53	bilisim	5213	1002	3	
5	13001	Murat	i	28	test	6753	1002	4	
6	13002	Beyza	l	36	bilisim	7123	1002	5	
7	13003	Efe	m	32	test	5891	1000	6	
8	13004	Kübra	n	29	üretim	6546	1000	7	
9	13005	Fatih	t	30	test	7005	1001	8	
10	13006	Kerim	s	41	üretim	7564	1001	9	
11	13007	Sıla	y	35	bilisim	6943	1003	10	
12	13008	Dilan	z	45	üretim	5963	1003	11	
13	13009	Emre	x	35	üretim	7591	-	12	
14	12130	Dilara	Patır	28	bilisim	12000	1004	messiCampeon	
15	12964	ceyda	patır	23	bilisim	5000.0	-	18	
16	13828	zeynep	doğan	34	Bilisim	3000.0	-	904	
17	13365	Ahmet	Uyan	54	Tasarım	4500.0	-	907	
18	13255	Ahmet	Yılmaz	54	Üretim	4500.0	1004	956	
19	13145	Canan	Öztürk	44	Üretim	4500.0	1004	954	
20	13348	Mesut	Yılmaz	45	Bilisim	3000.0	-	683	

FEmployee.txt después del cambio

Después de realizar este cambio ejecuta de nuevo la aplicación e introduce las credenciales de acceso:

- Company ID: **12130**
- Company Password: **messiCampeon**

Haciendo uso de la interfaz gráfica de **Netbeans** debes subir a tu cuenta de **github** el proyecto que has clonado en este ejercicio. La subida a **github** debe contener la modificación que has realizado en el fichero **FEmployee.txt** del proyecto.

Debes realizar capturas de pantalla en las que se muestren:

- Subida del proyecto al repositorio remoto en **github** desde **Netbeans**.
- Repositorio **github** con el proyecto subido.

(C) (4 pts) Descargar desde aules el fichero **ejercicio3_git.zip** dentro del cual encontraras los ficheros necesarios para realizar este ejercicio.

Debes realizar las siguientes acciones haciendo uso de la herramienta **git**:

1. Crea un directorio de nombre **repoEjercicio3**.
2. Inicializa un repositorio **git** dentro del directorio creado en el punto anterior.
3. Añade a git los 4 ficheros descargados desde aules y crea tu primer commit con el mensaje “primer commit”
(captura de pantalla 1 (de los puntos 1, 2 y 3))
4. Te has dado cuenta que te has equivocado en el mensaje del commit anterior y deseas modificarlo por el siguiente mensaje “primer commit de mi proyecto web”. **(captura de pantalla 2)**
5. Edita el fichero **index.html** añadiendo el siguiente contenido entre la etiqueta `<body>` `</body>`

```
<h1 class="titulo">Bienvenidos</h1>
```

6. Agrega los cambios realizados en el punto anterior a git y realiza un commit con el mensaje “Agregado encabezado nivel 1”
7. Edita el fichero **style.css** añadiendo al final del fichero el siguiente contenido:

```
body{
    background-color: darkseagreen;
}
h1.titulo{
    background-color: crimson;
    color: white;
    border: 1px solid yellowgreen;
}
```

8. Agrega los cambios realizados en el punto anterior a git y realiza un commit con el mensaje “Agregado color de fondo y estilos del titulo”
9. Ejecuta el comando `git log --all --oneline --graph` **(captura de pantalla 3)**
10. Ahora mismo debes tener 3 commits en tu historial. Sucede que no te han gustado los cambios añadidos en el último commit(punto 8) y quieres eliminarlo del historial de tu proyecto git, es decir, te quieres quedar con los 2 primeros commits y descartar el commit realizado en el punto 8. **(captura de pantalla 4)**

11. Edita el fichero **style.css** añadiendo al final del fichero el siguiente contenido:

```
body{
    background-color: #3a3d47;
}
h1.titulo{
    background-color: #1551a1;
    color: white;
    border: 1px solid yellow;
}
```

12. Agrega los cambios realizados en el punto anterior a git y realiza un commit con el mensaje "Agregado nuevo color de fondo y estilos del encabezado h1".

13. Crea 1 rama de nombre **branchAntonio** y crea una nueva rama de nombre **branchSara**.

14. Cambiate a la rama **branchAntonio** y edita el fichero **index.html** agregando el siguiente contenido después de la etiqueta `<h1></h1>`:

```

```

15. Agrega los cambios realizados en el punto anterior a git y realiza un commit con el mensaje "Agregada imagen turistica Nueva Zelanda"

16. Edita el fichero **style.css**:

- añadiendo al final del fichero el siguiente contenido:

```
img.nz{
    width: 100%;
}
```

- modifica la línea donde aparece: `background-color: #3a3d47;`
lo modificas por esta línea: `background-color: #da924a;`

17. Agrega los cambios realizados en el punto anterior a git y realiza un commit con el mensaje "Modificado el ancho de la imagen al 100% y cambio color de fondo"

18. Ejecuta el comando `git log --all --oneline --graph` **(captura de pantalla 5)**

19. Ejecuta el comando que muestra las diferencias entre el contenido del fichero `index.html` en el último commit y el contenido de ese fichero 2 commits atrás. **(captura de pantalla 6)**

20. Cambiate a la rama **branchSara** y edita el fichero **index.html** agregando el siguiente contenido después de la etiqueta `<h1></h1>`:

```

```

21. Edita el fichero **style.css**:

- modifica la línea donde aparece: `background-color: #3a3d47;`
lo modificas por esta línea: `background-color: aquamarine;`

22. Agrega los cambios realizados en los puntos 20 y 21 y realiza un commit con el mensaje "Agregada bandera de Nueva Zelanda y cambio color de fondo a aquamarine"

23. Ejecuta el comando `git log --all --oneline --graph` *(captura de pantalla 7)*
24. Cambiate a la rama **master** y realiza el **merge** con la rama **branchAntonio**. *(captura de pantalla 8)*
25. Estando situado en la rama **master** realiza el **merge** con la rama **branchSara**. En caso de que aparezca un conflicto resuélvelo para poder finalizar la operación de merge *(captura de pantalla 9)*
26. Elimina de git las ramas **branchAntonio** y **branchSara**.
27. Ejecuta el comando `git log --all --oneline --graph` *(captura de pantalla 10)*
28. Accede a tu cuenta de **gitHub** y crea un nuevo repositorio vacío con el nombre **repoRemotoGit**. *(captura de pantalla 11)*
29. Desde la consola de **git** ejecuta los comandos necesarios para subir el repositorio local que has creado en tu máquina al repositorio remoto que has creado en **gitHub** en el punto anterior. *(captura de pantalla 12)*

SUBIR A AULES:

- **Fichero `capturasEjercicio1Git.zip`, `capturasEjercicio2Git.zip` y `capturasEjercicio3Git.zip` con todas las capturas de pantalla.**
- **Fichero `repoEjercicio3.zip` el cual tendrá en su interior el directorio que contiene el repositorio local git del ejercicio 3.**
- **En la tarea de aules del examen existe un cuadro de texto en el cual debéis introducir las 3 URLs de los repositorios remotos en gitHub que habéis creado en los 3 ejercicios de GIT.**