

Acceso a Datos – Tarea 06

Desarrollo de aplicaciones multiplataforma

Felipe Rodriguez

Contenido

[Ejercicio 1 3](#_Toc511329284)

[Añadir tabla de calificaciones 3](#_Toc511329285)

[Creando el componente 4](#_Toc511329286)

[Atributos 4](#_Toc511329287)

[Constructor 4](#_Toc511329288)

[Métodos getter y setter 5](#_Toc511329289)

[Clase auxiliar Calificación 5](#_Toc511329290)

[Atributo Calificaciones 6](#_Toc511329291)

[Atributo Tamanio y método getTamanio 6](#_Toc511329292)

[Método recargarFilas 7](#_Toc511329293)

[Atributos de conexión a base de datos 7](#_Toc511329294)

[Método seleccionar fila 8](#_Toc511329295)

[Método seleccionar DNI 8](#_Toc511329296)

[Añadiendo eventos 9](#_Toc511329297)

[Método agregarNotaModulo 9](#_Toc511329298)

[Método modificarNotaModulo 10](#_Toc511329299)

[Oyentes de los eventos 10](#_Toc511329300)

[Implementando el componente 11](#_Toc511329301)

[Sobre rutas del archivo .jar 11](#_Toc511329302)

[Creando la clase de conexión AccedeDB 11](#_Toc511329303)

[Método listado 11](#_Toc511329304)

[Método añade 11](#_Toc511329305)

[Método modifica 12](#_Toc511329306)

[Métodos de los eventos 12](#_Toc511329307)

[Utilizando el componente 12](#_Toc511329308)

[Incluyendo los métodos en el método main 12](#_Toc511329309)

[Visualizando los resultados 13](#_Toc511329310)

[Ejercicio 2 14](#_Toc511329311)

[Conclusiones personales de la tarea 6 14](#_Toc511329312)

[Posibles mejoras en tu programa y en el planteamiento/enunciado de la tarea 14](#_Toc511329313)

[Dificultades que te has encontrado en la realización de la práctica 14](#_Toc511329314)

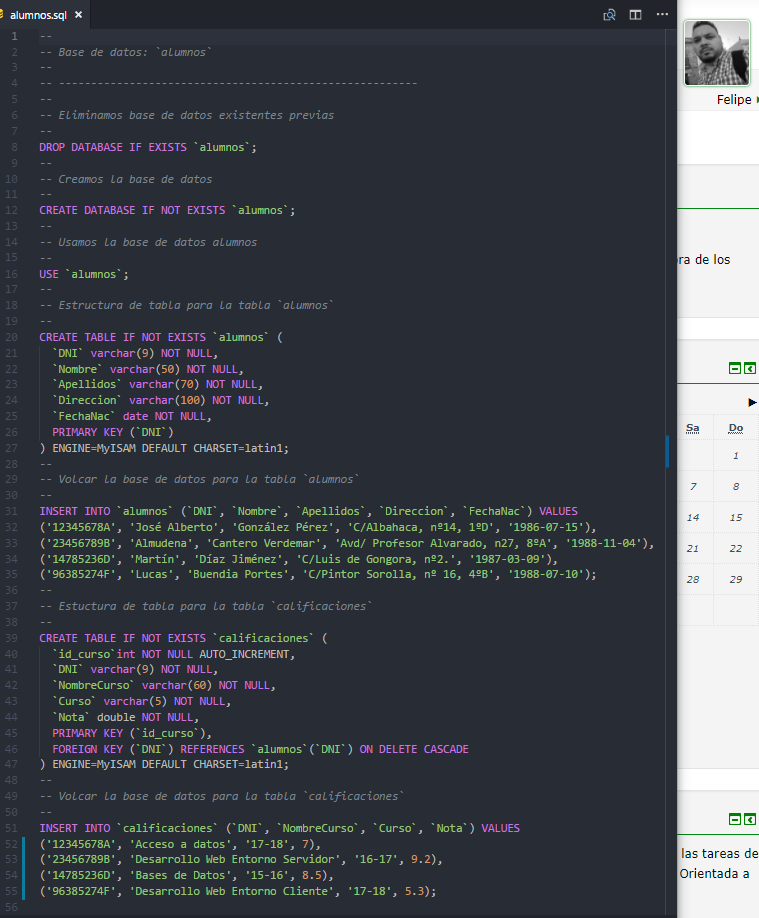
[Opinión personal del trabajo realizado, las herramientas, tecnologías usadas, etc. 14](#_Toc511329315)

|  |
| --- |
| ***Nota del Alumno sobre la tarea:***  *Se han realizado diversos cambios y mejoras en la estructura del ejercicio de ejemplo del temario del que se ha partido como base para la tarea, siendo principalmente los que continúan, que serán mostrados en detalle más adelante:*   * *Se han realizado diversas mejoras a la base de datos del ejemplo, que se adjunta a la presente tarea, aunque el ejercicio es completamente funcional y compatible con la tabla de calificaciones normal.* * *No es necesario copiar el .jar, dado que en el apartado 2 donde hacemos uso del componente, hace llamada a la ruta relativa a su ubicación, por lo que sólo tendremos que hacer clean and build cada vez que hagamos un cambio y encontrarse los dos proyectos en la misma carpeta.* * *Este ejercicio ha sido realizado con una base de datos ubicada en red local, por lo que se ha modificado el ComponenteBean con una serie de atributos finales que pueden ser modificados con la configuración del servidor local/red así como su usuario y pass correspondiente, no teniendo que ser modificado posteriormente en el resto de métodos.* * *Se han realizado algunos cambios en el funcionamiento de ciertos métodos, dado que no funcionaban correctamente. En el método seleccionaDNI(), al introducir un DNI, entraba en bucle infinito y no terminaba la ejecución. En el método recargarFilas(), se ha implementado limpiar el vector al recargar, dado que al no ser limpiado el vector, se añadían a los existentes la nueva lista de filas cada vez que se procedia a la recarga.* * *Se ha añadido el método getTamanio(), que devuelve el tamaño en ese instante del vector de calificaciones, dado que en la estructura del ejemplo, solo permite listar si conocemos el número de filas que posee el vector. De esta manera, siempre se listarán todas sin tener que conocer el número de filas.* |

# Ejercicio 1

## Añadir tabla de calificaciones

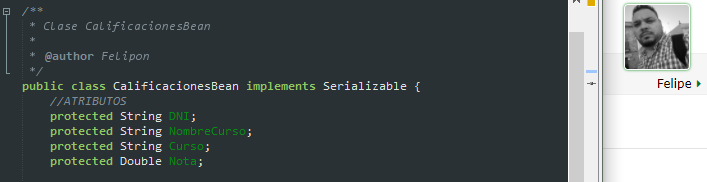
Se crea la el código SQL de la base de datos, ampliándose la existente del ejemplo:



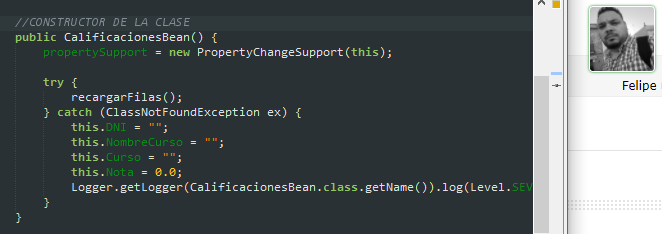
|  |
| --- |
| ***Observaciones:***   * Se ha incluido que al importar el script SQL, elimine previamente alguna versión existente de la base de datos para que sea sustituida por esta misma. No es necesario, dado que en principio la tarea esta realizada con la tabla normal y corriente. * Hay que modificar en el componente la url, user y Password de conexión a la base de datos previamente si queremos que funcione. * El campo DNI de *calificaciones* será clave foránea del campo DNI de la tabla *alumnos* añadiendo integridad referencial. * Para la consecución de lo anterior, se ha creado el campo id\_curso como clave principal para poder usar DNI como clave foránea. |

## Creando el componente

### Atributos

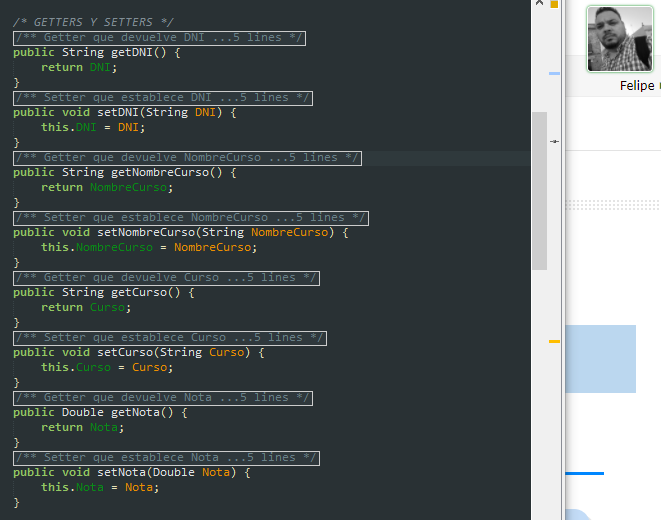


### Constructor

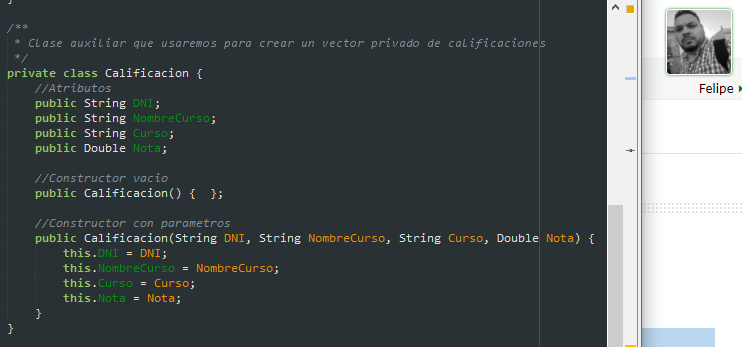


Al ser instanciado, llamará al método recargarFilas(); que volcará en un vector los valores de la tabla.

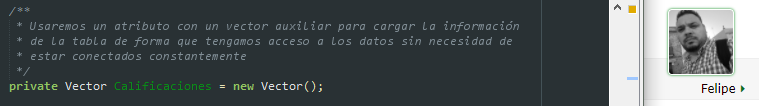
### Métodos getter y setter



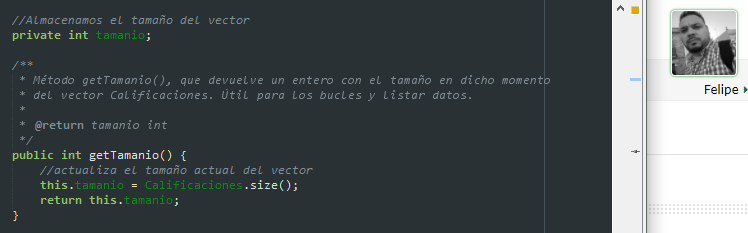
### Clase auxiliar Calificación



### Atributo Calificaciones

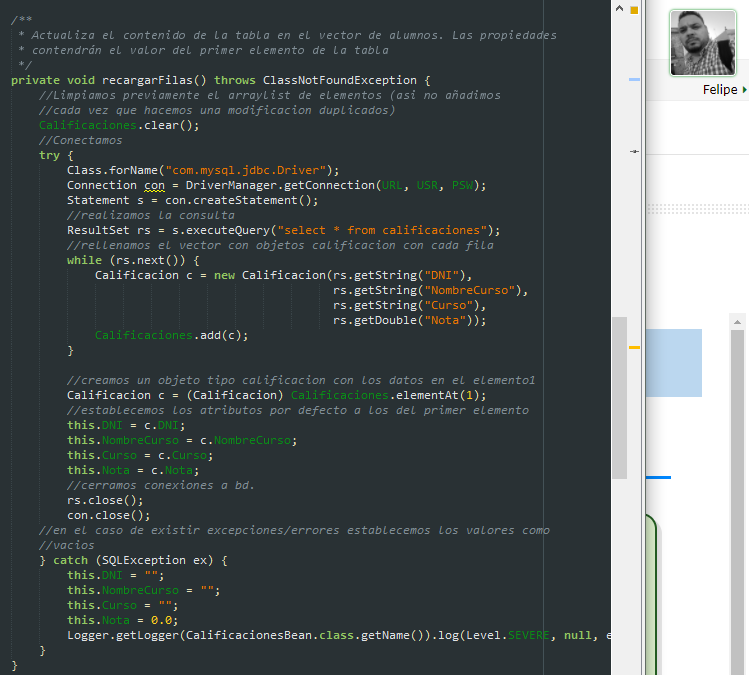


### Atributo Tamanio y método getTamanio



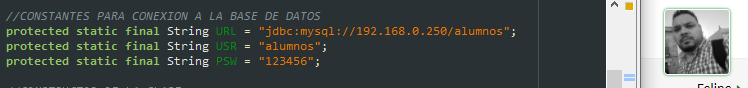
|  |
| --- |
| Este método se ha implementado para recorrer el vector Calificaciones y poder mostrar correctamente todas las calificaciones al proceder a su listado. En el ejemplo de la unidad debía conocerse el número finito de filas existentes del vector.  Ahora es posible listar los cursos sin necesidad de conocer el tamaño del vector interno. |

### Método recargarFilas



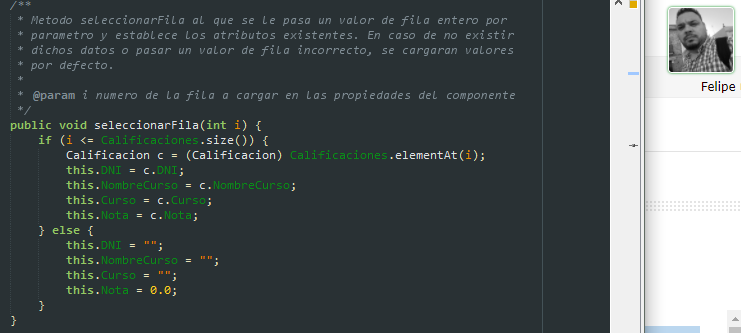
|  |
| --- |
| Se ha modificado el método para que limpie el vector previamente antes de recargar el contenido de filas en el vector. En el caso de no realizar este paso, se insertarán al final del vector existente, por lo que creará un número irreal con respecto a la tabla, mostrando registros duplicados. |

### Atributos de conexión a base de datos

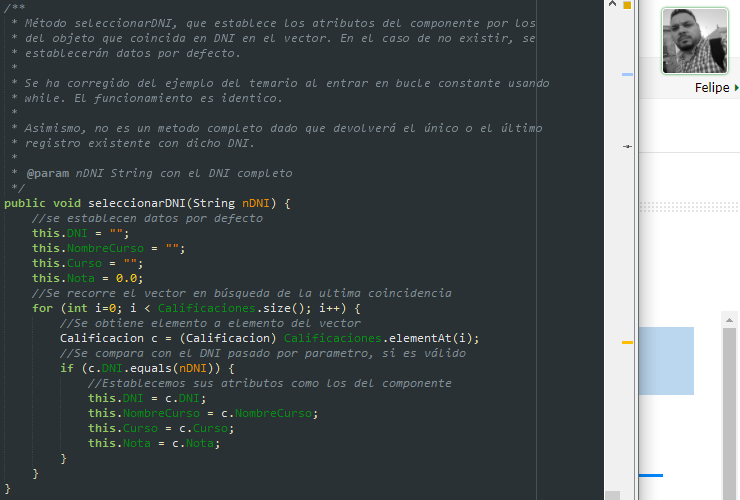


|  |
| --- |
| Se han creado estos atributos al inicio de la clase con el fin de facilitar la configuración de la conexión a la base de datos como atributos constantes (finales). De esta manera se pueden evitar posibles errores. Están modificados para conectar a un servidor de BD en red local con usuario y clave concreta, verificar estos datos previamente antes de ejecutar el proyecto. |

### Método seleccionar fila



### Método seleccionar DNI

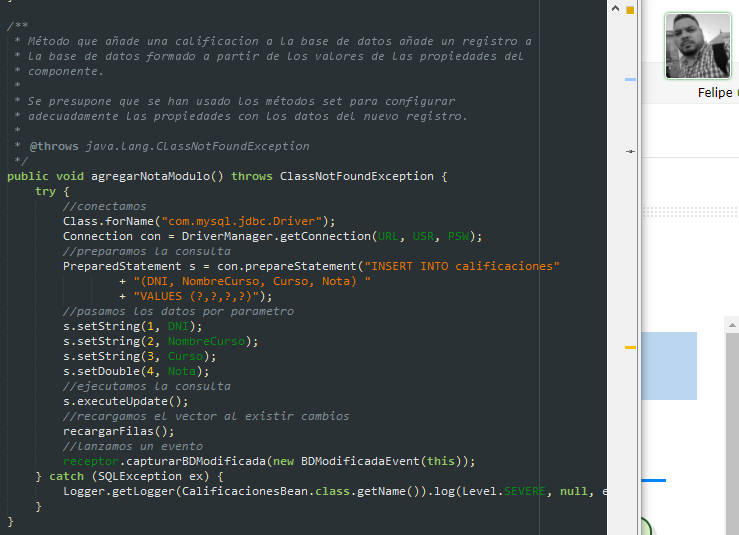


|  |
| --- |
| Se ha modificado el while por un ciclo for (probado, no terminaba de funcionar, quedando en bucle continuo). |

### Añadiendo eventos



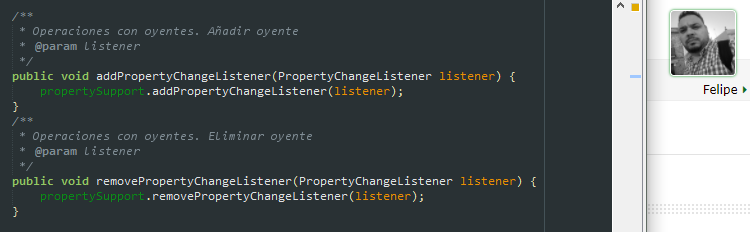
### Método agregarNotaModulo



### Método modificarNotaModulo



### Oyentes de los eventos

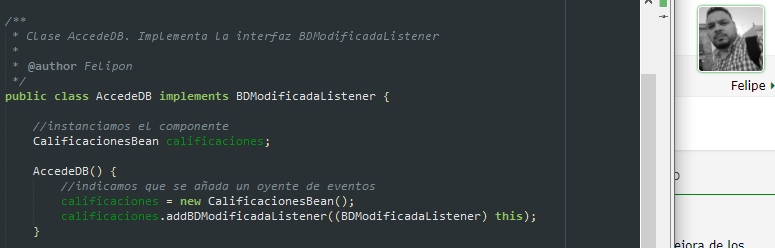


## Implementando el componente

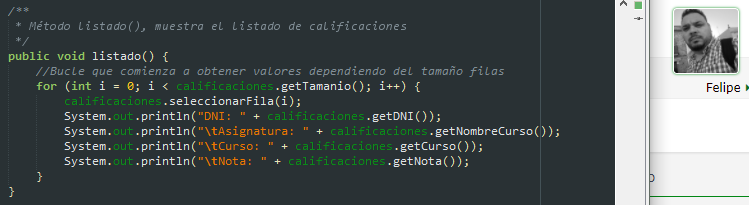
### Sobre rutas del archivo .jar

|  |
| --- |
| El archivo .jar se ha mantenido en su ubicación original. Se ha añadido al proyecto con ruta relativas, asi cada vez que se haga *clean and build* en el proyecto1 estará actualizado en el proyecto2 siempre que se mantengan los nombre de los directorios inalterados y juntos. |

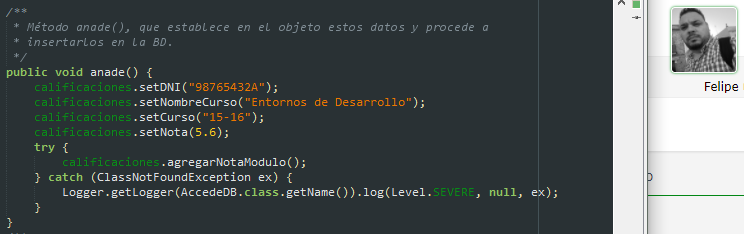
### Creando la clase de conexión AccedeDB



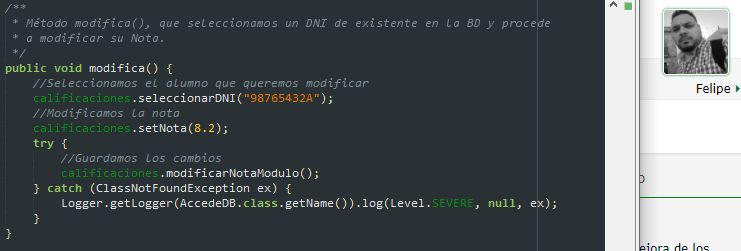
### Método listado



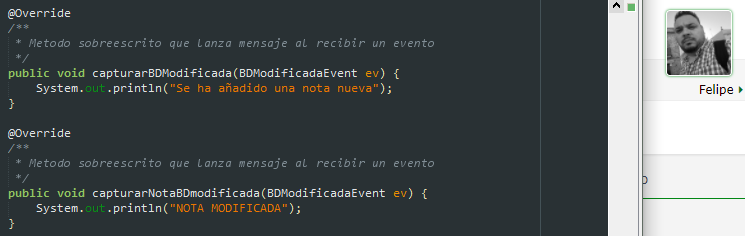
### Método añade



### Método modifica

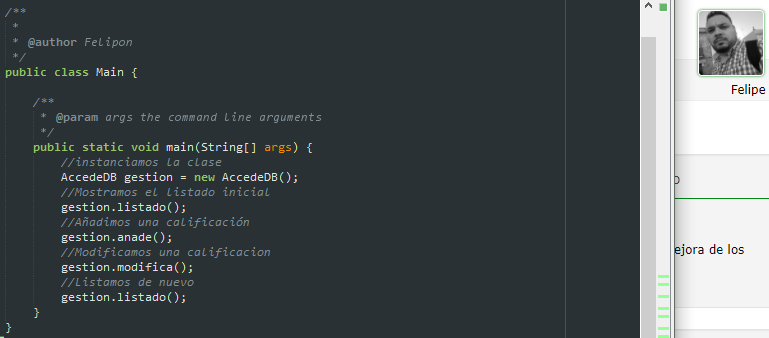


### Métodos de los eventos

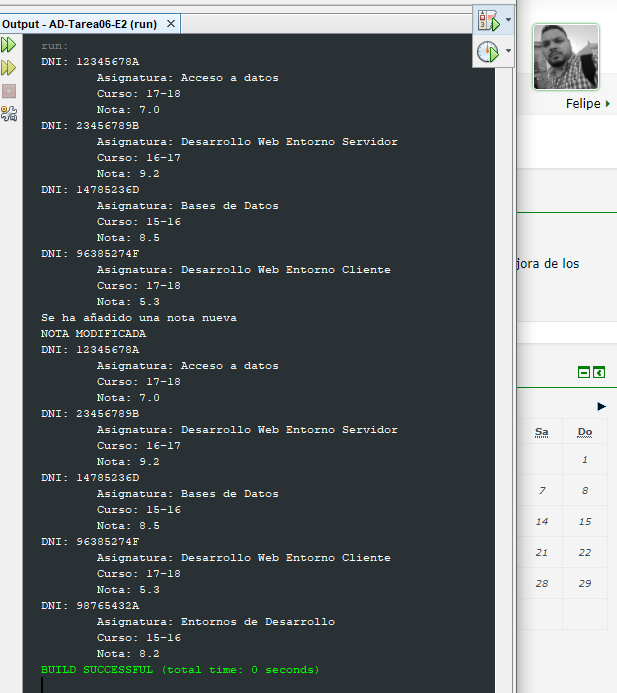


## Utilizando el componente

### Incluyendo los métodos en el método main



## Visualizando los resultados



# Ejercicio 2

## Conclusiones personales de la tarea 6

### Posibles mejoras en tu programa y en el planteamiento/enunciado de la tarea

Como posible mejora, pues la que no falta nunca, una interfaz GUI, pero creo que podría restarle tiempo al ejercicio. Sobre el planteamiento, similar al ejercicio de ejemplo, pero salteado con algunas cosas más.

### Dificultades que te has encontrado en la realización de la práctica

Como dificultad he encontrado el temario, esta vez me ha parecido lioso el planteamiento del temario y lo “escuetillo” que era en ciertas partes. A base de machacar el ejemplo a prueba y falla he logrado sacar la tarea (o al menos eso creo).

No he terminado de pillarle el tema a los eventos, ciertas partes no se para que se realizan, ni por qué funcionan, sólo sé que están ahí y que todo el código en conjunto funciona. Tendré que explorar ese tema mejor.

### Opinión personal del trabajo realizado, las herramientas, tecnologías usadas, etc.

Me ha gustado esta materia y conocer los eventos. Sabía como se implementaba el modelo MVC de forma web, pero desconocía que podía implementarse aquí también.

Me gustaría aclararme con los eventos y ciertos aspectos de la creación de componentes, ya las preguntaré en el foro.

