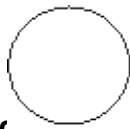


Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Para escribir el programa utilizare un patrón arquitectónico llamado MVC que significa Model View Controles por sus siglas en ingles, este patrón recomienda construir cualquier programa de computo utilizando una arquitectura basa en tres capas.



La capa **Modelo** (M):

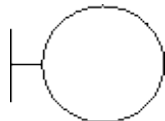
Es la encargada de producir la los datos o la información que el usuario necesita y la entrega a la capa de Control, para que sea entregada a su vez a la capa de vistas, existe una técnica llamada databinding, que permite al Modelo notificar a la capa de Vistas cuando ocurre un cambio en su estado (cuando se producen cambios en el Modelo, ya sea por inserción, actualización, eliminación o producción de datos), una vez notificada la Vista sobre el cambio, esta actualiza la información que presenta sobre dicha parte del modelo que corresponde ese cambio.

El modelo representa las líneas de código que dan solución a la lógica a desea sistematizar, en un programa basado en metodología Orientada a Objetos, estas líneas de código están representadas por clases, las cuales permiten construir objetos que se relacionan entre sí, intercambiando mensajes uno al otro, en la capa del Modelo se colocan las clases que contienen la lógica necesaria para realizar las actividades de persistencia CRUD (Create, Read, Uopdate o Delete), cálculos propios de del problema en cuestión (compras, ventas, nomina, impuestos pedidos, intereses, prestamos, proyectos, contabilidad, etc).

El nombre Modelo le fue dado a esta capa precisamente porque el código que aquí se encuentra, debe ser una fiel representación o Modelo de las operaciones que se desean asistir mediante software.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Al iniciar el proceso de construcción del Software, la capa de Modelo es la primera fase que se debe tener bien clara y definida, por esta capara se inicia el proceso de análisis diseño y programación, por esta razón nombre del patrón inicia con M (Model), se trata de transformar la solución problema a líneas de código Orientado a Objetos en nuestro, caso a líneas de código Java.



La capa de Vistas (V):

Se le llama capa de Vistas por que contiene toda la lógica de código fuente utilizada para interactuar con los usuarios del software, la palabra vistas hace referencia a las Interfaces que el Usuario ve, que le permiten de una u otra forma intercambiar información con el software, como por ejemplo llenar un formulario, pulsar un botón, mover o arrastrar un elemento grafico, seleccionar una opción de menú o un ítem de alguna lista de datos, también se utiliza la capa de vistas para presentar la información que el sistema procesa, como producto de la ejecución de alguna de las operaciones definidas en la capa de Modelo, que una vez ejecutadas generan la información necesaria por el usuario, se entrega a la capa de Vista en un formato propio del software, por ejemplo mediante colecciones de Objetos, entonces la Vista se encarga de aplicar un formato a dichos objetos, de tal forma la información contenida en ellos pueda ser presentada al usuario de una forma clara, explicita y en un idioma reconocido y comprendido por ellos.

Una vez esté finalizada la capa de Modelo, el próximo paso es definir y construir la capa de Vistas, ya que existe una fuerte relación entre la forma y diseño de las vistas y las propiedades y operaciones de los objetos que estructuran al modelo, por este motivo es la segunda fase del nombre del patrón Model View. d

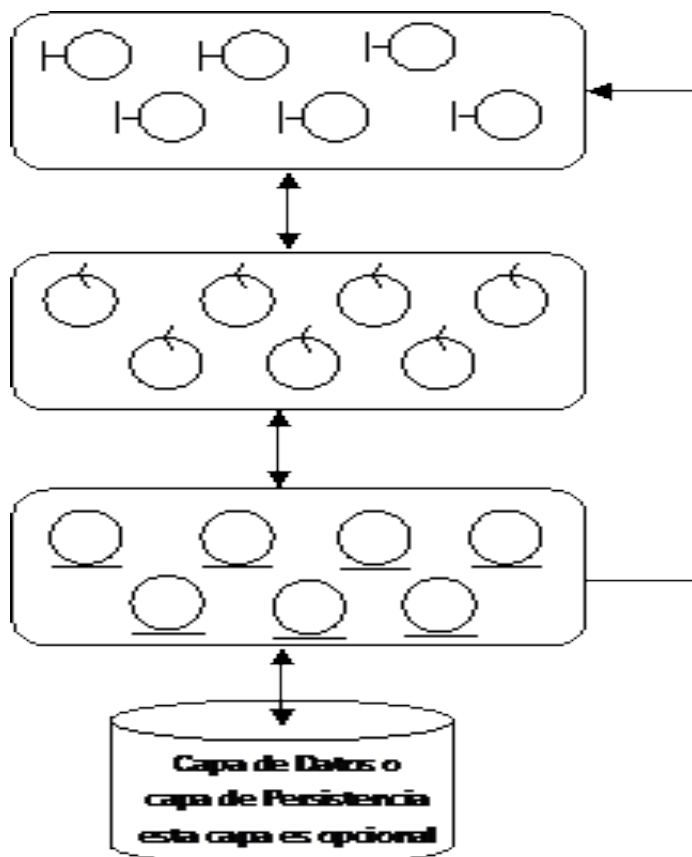
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

La capa de  C):

Esta capa como su nombre lo indica contiene todo el código fuente necesario para controlar las acciones (eventos y solicitudes) que los usuario realizan desde la capa Vistas, entonces las direcciona al respectivo elemento de la capa de Modelo responsable de ejecutar o realizar la operación o solicitud requerida, el control espera la respuesta del Modelo y redirección dicha respuesta a la parte de la Vista encargada de presentar la información de correspondiente a la respuesta enviada por el Modelo.

El funcionamiento básico del patrón MVC se puede apreciar en si siguiente figura

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



1. Las Vistas interactúan con los usuarios.
2. Reciben solicitudes para que el Modelo ejecute cierta operación, o implemente reciben datos de los usuarios, esta información es recibida desde el exterior mediante algún tipo de interfaz, que puede ser Voz, Texto desde el teclado o desde otro sistema, Movimientos del Ratón o de otro dispositivo de entrada, captura de



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

tacto digital sobre un plano, etc..

- Las Vistas verifican que el formato de esos datos sea el adecuado para entregarlo al los Controladores, es decir que la información cumpla con los mínimos requerimientos de formato que se indican en el modelo (que sea numérica, o solo texto, que tenga o no tenga al menos x numero de caracteres, que X datos sea obligatorio, etc), si esto no se cumple, las Vistas notifican al usuario sobre la violación del formato ocurrida.
- Una vez verificado que el formato de los datos sea el correcto, entonces las Vistas envían los datos a los Controladores para que esto se encarguen de decidir cual elemento del Modelo es el responsable de procesar dichos datos, antes Cada Controlador se decide si debe aplicar otro formato a los datos para poderlo entregar a la Modelo.
- Los elementos de la capa de Modelo responsables de procesar los datos los reciben e igualmente verifican que el formato de los datos sea el adecuando.
- Realizan algún tipo de operación con estos datos (generan información o realizan acciones necesarias), estas operaciones pueden ser dirigidas a la capa opcional de Datos o Persistencia, para esta ultima que realice operaciones de almacenamiento, modificación, eliminación o consulta sobre dichos datos, a los datos se les puede dar otro tratamiento como por ejemplo enviarlos por la red a otro software o sistema informático para que realice las acciones pertinentes sobre los mismos, o simplemente cualquier numero de el modelo realiza cálculos necesarios con estos datos.
- Sea cual sea el tratamiento que se le aplique a los datos, los elementos de la capa



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

de Modelo siempre deben entregar una respuesta a la capa de Control para finalizar con su espera.

8. Los controladores vinculados con la respuesta nuevamente verifican el formato de los datos, analizando si la información entregada por la capa de Modelo tiene o no tiene el formato adecuado para ser entregada a la capa de Vistas, y esta pueda iniciar con la presentación de dicha información, si se requiere la capa de control aplicara el formato necesario.
9. Por último la capa de Vistas recibe la información proveniente de la capa de Control, que a su vez la recibe de la capa de Modelo, la prepara para ser presentada al usuario en el formato en que este la haya solicitado.
10. Existe la posibilidad de que los datos en la capa de Modelo cambien su estado (se inserten, Modifique o eliminen), o simplemente puede ocurrir que el Modelo realice alguna operación cuyo resultado genere información importante para el usuario, entonces, la capa del Modelo decide notificar a las Vistas vinculadas con dicha información, informándoles sobre las novedades.
11. La capa de Vistas recibe la notificación emitida directamente por la capa de Modelo y procede a Actualizar la presentación de la información conforme se le haya entregado esta notificación.

Es de anotar que los elementos pertenecientes a una misma capa pueden comunicarse unos con otros, es decir una Vista puede comunicarse con otra Vista, un Controlador con otro Controlador y elemento del Modelo con otro elemento perteneciente al Modelo.

Se pueden comunicar elementos pertenecientes a diferentes capas, pero solo siguiendo esta



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

simple regla:

De la capa Vistas a la Capa Control y viceversa

De la capa Control a la Capa Modelo y viceversa.

De la capa Modelo a la Capa Vista, pero debe existir un mecanismo intermedio que realice la conexión entre ambos, ya que el Modelo nunca debe saber a qué elemento exacto de la Vista debe Notificar, de hecho el Modelo no sabe de la existencia exacta de los elementos de la Vista.

Nunca debe haber conexión directa entre la capa Vista a la capa Modelo.


Ventajas de patrón MVC:

Permite organizar el código fuente en capas según su objetivo en el software.

Separa claramente el código de Interfaz de Usuario del código de Lógica del programa (Negocio)

Permite implementar varios tipos de interfaz de Usuario “Vistas” (Consola y/o Grafica para Escritorio, Voz, Web para internet, táctil para sistemas empujados, Móvil para dispositivos inalámbricos, etc.) sin cambiar una sola línea de código del Modelo (el negocio o lógica del programa).

Facilita el mantenimiento del software, permitiendo identificar fácilmente los elementos donde hay que hacer correcciones o mejoras, sin tener que leer gigantescas de líneas de código espagueti (código de Vistas combinado con código de Lógica de negocio).

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena		PIENSA EN GRANDE Piensa en UNISINÚ
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas	
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27	
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI	
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 		

Desventajas del patrón MVC:

Primero se debe tener un dominio completo de la lógica del programa (Negocio), antes de iniciar con la construcción del programa, debido a que este comienza a desarrollarse pero la capa de Modelo.


Al comienzo es muy difícil de entender e implementar.

Si no se tiene bien clara e identificadas las funciones de la capa de Control, se puede prescindir de ella y realizar un sistema de dos capas VM (Vista-Modelo), escribiendo código espagueti (Mezcla de Vista con Modelo), y se termina por integrar demasiado las dos capas, haciéndolas dependientes una de otra.

Se debe hacer mucho análisis en los diferentes elementos que estructuran el software, para poder definir muy bien las líneas fronterizas entre cada una de las tres capas del patrón MVC.

El patrón de diseño de software Observador y Observado

Los patrones de diseño de software son técnicas de programación, pensadas y diseñadas con el fin de ofrecer la mejor solución para ciertos problemas cotidianos que se presentan durante la fase de diseño y programación de software Orientado a Objetos (DyPOO), estos patrones han sido puestos a prueba en infinidad de proyectos de software, a tal punto que se han convertido en estándar para el diseño de software OO y cada uno de ellos está firmemente vinculado a solucionar un problema en particular, es decir no existen patrones de diseño genérico, por el contrario están clasificados según el tipo de problemas al que dan solución.

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena		PIENSA EN GRANDE Piensa en UNISINÚ
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas	
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27	
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI	
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 		

Existen patrones de diseño para:

- La creación problemas relacionados con la creación de objetos.
- La estructura de Orientada a objetos de un programa
- El comportamiento de los Objetos que conforman el programa.

Uno de los patrones mas utilizados para implementar el patrón de arquitectura MVC es el patrón de diseño de comportamiento Observador y Observado.

El **Observer & Observable** consiste en hacer que el estado (valor de las variables, atributos o propiedades) de un objeto productor o manejador de datos (característica principal de un objeto de la capa de Modelo) sea vigilado u observado (Observable) por otros objetos Observadores (Observers), a los que les interesa ser notificados cada vez que cambie el estado del objeto Observado.

Un objeto observado (Observable) posee una lista de su objetos Observadores (Observers), los cuales pueden ser agregado o removidos de la lista en tiempo de ejecución.

El objeto Observado debe poseer mecanismos que le permitan:

- Notificar a todos o a uno en particular de sus objetos Observadores sobre los cambios ocurridos en su estado (valores de sus variables).
- Agregar cualquier cantidad de Observadores en tiempo de ejecución a su lista de Observadores



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

- Remover todos o cualquier cantidad de Observadores de su lista de Observadores
- Buscar un Observador específico en su lista de Observadores
- Saber cuántos Observadores tiene en su lista en cualquier momento deseado

El objeto Observador poseer mecanismos que le permitan:

- Consultar el nuevo estado del objeto Observado de interés
- Verificar si el cambio de estado se efectuó en el objeto Observado de interés y no en otro de origen desconocido.
- Verificar que la notificación fue realizada por el objeto Observado de interés.
- Actualizarse con la información exacta del cambio de estado exacto.

Este patrón de diseño al igual que muchos otros patrones de diseño ya se encuentra programado en muchos de los lenguajes de programación Orientados Objetos, solo tenemos que conocer las librerías y clases que los conforman, para posteriormente utilizarlos en nuestros proyectos.

Patrón de diseño Observador y Observado en Java

Java es uno de los lenguajes de programación que más se ha utilizado para documentar e

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

implementar patrones de diseño de software, en el caso del patrón Observador y Observado Java posee dos elementos en su librería **java.util** del JSDK, el primero es una Interfaz Java llamada **Observer** y la segunda es una clase llamada **Observable**.

La interfaz **Observer**

Observer
<code>+update(Observable o, Object arg): void</code>

Declara un único método o función llamado **update(Observable obs, Object arg)**, esta función debe ser implementada por clases concretas que deseen comportarse como Observadores de algún objeto en particular, update recibe dos parámetros el primero es el objeto Observado que realizo la notificación sobre el cambio de su estado, el segundo es el nuevo valor del estado que ha cambiado.

Este método se utiliza para realizar operaciones concernientes a la actualización de otros objetos cada vez que cambien el estado en el objeto observado.

La clase **Observable**


Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Observable
<pre> +notifyObservers(): void +notifyObservers(Object o): void +addObserver(Observer o): void #clearChanged(): void +countObservers(): void +deleteObserver(Observer o): void +deleteObservers(): void +hasChanged(): boolean +setChanged(): void </pre>

Esta clase abstracta se utiliza para ser heredada por los objetos que deseen comportarse como objetos Observados, esta clase contiene toda la funcionalidad necesaria para agregar, quitar, buscar, contar y notificar objetos Observadores de la lista de objetos del objeto que desea ser observado.

Los pasos para utilizarlas son:

1. Crear al menos una clase Observada que **herede** de la clase **Observable**.
2. Identificar los atributos del objeto observado, para los que se desea agregar objetos Observadores (Observer), que serán notificados automáticamente justo cuando se efectuó el cambio de valor (estado) de dichos atributos (variables).
3. Sobre escribir los métodos de la clase Observable si se desea que al menos uno de ellos ejecute algún comportamiento especial.
4. Dentro de las funciones (métodos) que realizan los cambios (set) de sus correspondientes

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena		PIENSA EN GRANDE Piensa en UNISINÚ
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas	
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27	
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI	
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 		

atributos hacer la notificación de los objetos observadores, llamado a los métodos heredados **setChanged()**,

notifyObservers(), **notifyObservers(Object observador)**, **notifyObserver(Observer observador)**.

5. Crear al menos una clase Observadora que **implemente** de la interfaz **Observer**.
6. Definir (implementar) el código necesario para la función **update(Observable observado, Object valor)** declarada como parte de la interfaz Observer, este código se utilizara para obtener el nuevo valor de las variable que cambio su estado.
7. Verificar si el objeto Observado que se pasa como parámetro de la función update corresponde al objeto que se esta observando.
8. Registrar un objeto observado y luego otro observador
9. Registrar el objeto observador dentro de la lista de objetos Observadores del objeto observado creado con anterioridad, esto se hace utilizando la función **addObserver(Observer observador)**, del objeto observado.

Principio básico de Hilos (Threads)

Para poder generar información del tiempo, utilizara un proceso concurrente, también conocido como Hilo, que en java es representado por la clase **Thread** del paquete básico **lang** del JSDK.

Los hilos son fragmentos de código que al ser iniciados (llamando al método **start**) se ejecutan al mismo tiempo junto al programa que los inicia, es decir que su ejecución sale de la pila o cola de

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

operaciones a ejecutar secuencialmente, pasando a ser un proceso (programa) más en la lista de procesos del sistema operativo que compiten por el uso del procesador (CPU), esto les permite ejecutarse concurrentemente (en paralelo) fuera de la jerarquía de operaciones que deben ser ejecutadas por el programa que los inicio. A esto se le conoce como programación concurrente o Hebras (threads).


Los hilos se pueden iniciar (start o run), suspender o dormir (sleep) y detener (stop).

Dentro del de la función **run()** (operación o metodo) se escribe el código que se desea ejecutar de forma concurrente (paralela) a nuestro programa.

La función **strar()** se utiliza para llamar automáticamente a la función **run()**, es decir **start()** hace que comience a ejecutarse el código de forma concurrente.

La función **sleep(long tiempo)** permite que el hilo (código concurrente) se detenga por un tiempo, es decir el hilo dormirá durante el tiempo que nosotros le indiquemos en la función **sleep(tiempo)**, donde el tiempo se expresa en milisegundos, una vez finalizado el tiempo de latencia, el hilo volverá a ejecutarse de forma concurrente nuevamente, dormir a un hilo permite que el resto de hilos de nuestro programa (incluso el programa mismo es un hilo, es el hilo con mayor prioridad de uso del CPU) tengan mayores posibilidades de utilizar al CPU para que ejecute porciones del código definido en su función **run()**, esta característica permite que el programa sea equitativo en cuanto a la asignación de tiempos de uso del CPU para sus proceso concurrentes (Hilos o Threads), mejorando significativamente el desempeño del mismo.

La función **stop()** obliga o fuerza la terminación del hilo, es decir dice al CPU que no siga ejecutando por ningún motivo el código que se encuentra dentro de la función **run()** del hilo que se ha detenido, la especificación que indica cómo trabajar con hilos, desaconseja el uso de este función como mecanismo de finalización de un hilo, debido a que una vez detenido iniciado el hilo,

 <div> UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena </div> <div> PIENSA EN GRANDE Piensa en UNISINÚ </div>	
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

este entra fuerte competencia con el resto de procesos (otros hilos o programas que se están ejecutando en el S.O) y al momento de obligarlo a finalizar con el método **stop()** puede ocurrir que justo en ese momento dicho hilo este siendo procesado (atendido) por la CPU, algo que muy seguramente generaría un Error de Interrupción de proceso, error que se almacena en la clase Java InterruptedException del paquete básico lang del JSDK.

Como alternativa a la llamada de la función **stop()**, se puede escribir algún mecanismo dentro del código de la función **run()** que finalice su ejecución, lo mas común y fácil es mediante alguna condición de validación que condicione la ejecución del código que se encuentra dentro del método **run()**.

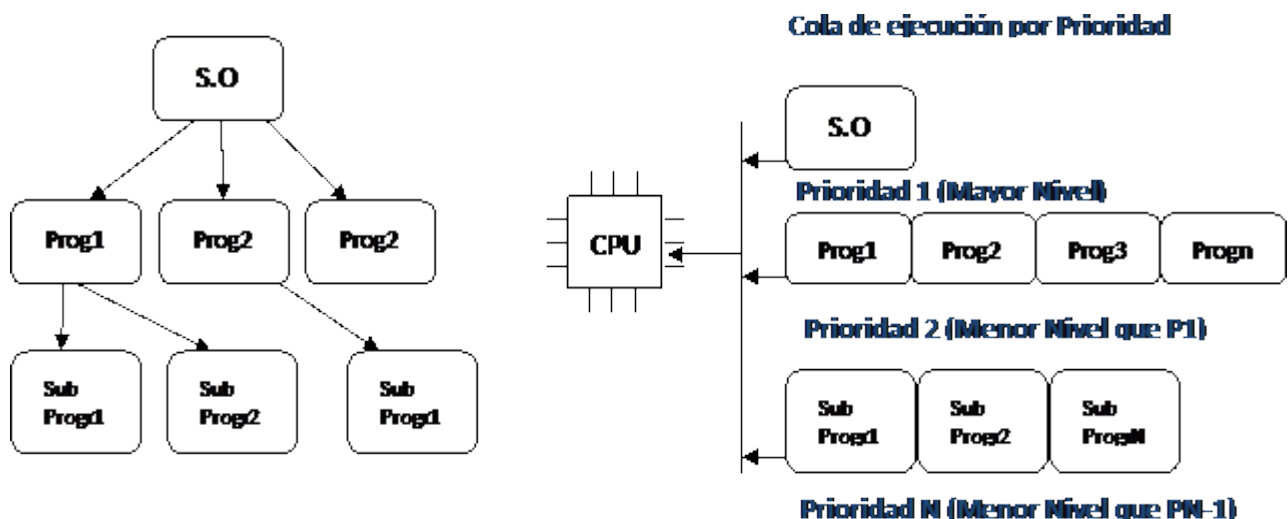
Un programa puede iniciar la ejecución de muchos hilos (programa o proceso que se ejecución de forma concurrente a otros programas del S.O), e incluso el programa que inicia la ejecución de otros hilos es también un hilo en ejecución, pero este programa matriz, programa iniciador o programa primario por defecto tiene mayor prioridad (mayor nivel de ejecución) que los programas Hilos que este ha iniciado, permitiendo finalizar el programa primario y al mismo tiempo finalizar de forma automática todos los hilo secundario (de menos prioridad) que se hayan iniciado por el programa primario que deseamos cerrar.

Todos los programas que se ejecutan dentro del el SO son por defecto hilos (procesos o subprocesos del SO) secundarios que fueron iniciados por el Sistema Operativo, algunos son ejecutados de forma autónoma por el mismo SO, pero otros son iniciados por el SO como respuesta a una solicitud hecha por nosotros los usuarios.

Sea cual sea la forma en que se inicia un programa (hilo o proceso o subproceso) dentro del SO, este tendrá menor prioridad de ejecución de la que posee el SO, de no ser así, el SO dejaría de funcionar simplemente porque existirían procesos con mayor prioridad de ejecución, negándole cualquier posibilidad de utilizar la CPU mientras ellos estén en ejecución.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entorno Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Los hilos (programas concurrentes o subprocesos de otro proceso indicador) compiten por pequeños tiempos de CPU con todos los procesos de su misma prioridad (Nivel de ejecución),



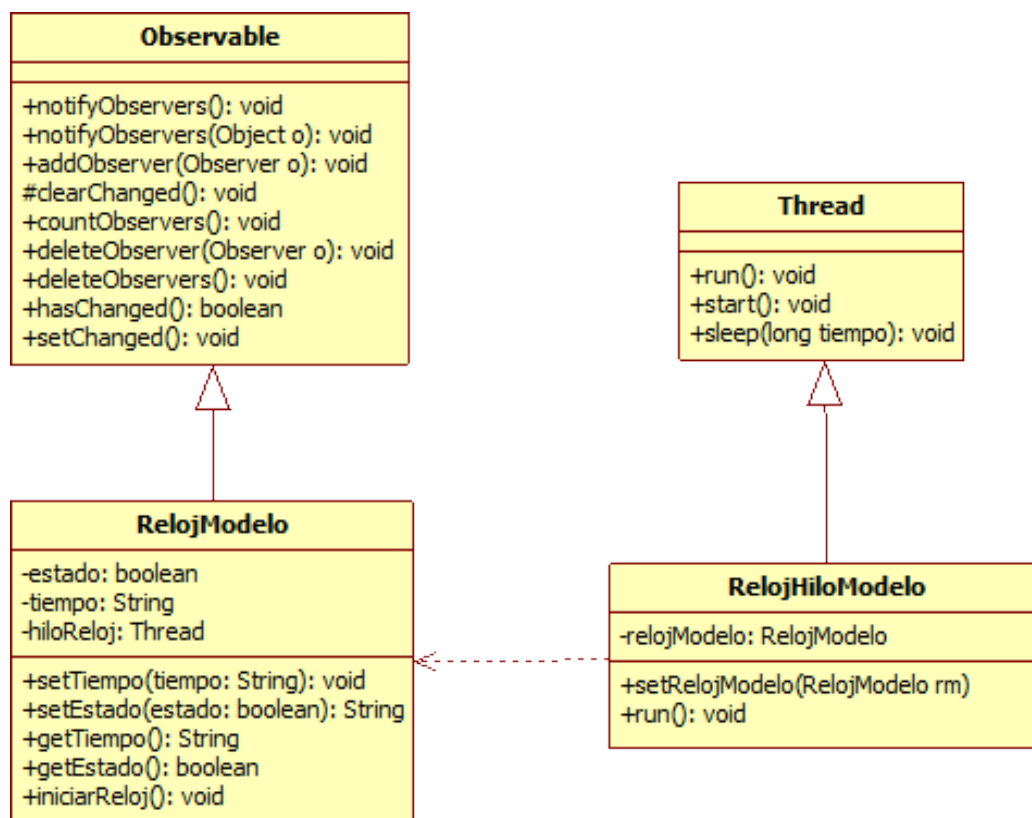
Es posible cambiar la prioridad de un Hilo para que aumente o disminuya su derecho (Nivel) de ejecutarse en el CPU, esto se puede hacer mediante la función **setPriority(int nivel)** de la clase Thread del JSDK, los niveles de prioridad van de 1 a 5, donde 1 es la prioridad de menor valor y 5 es la prioridad de mayor valor, Thread ofrece dos tres variables para manejar con mayor comodidad las prioridades de un hilo, se trata de MIN_PRIORITY, NORM_PRIORITY y MAX_PRIORITY, por ejemplo:

Diseño del Sistema



Una vez definida la arquitectura de paquetes del proyecto en función, lo siguiente es definir la arquitectura de clases que permitirán crear la estructura de objetos del sistema.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Las clases del Modelo



He diseñado dos clases para representar la lógica del modelo (Negocio), estas clases generan la información necesaria en el sistema, que en nuestro caso corresponde datos sobre el tiempo en formato hh:mm:ss a, esta información es producida por la clase **RelojHiloModelo**, quien hereda de **Thread** para poder crear un Hilo personalizado que de forma concurrente obtiene del computador datos sobre la hora, los minutos, los segundos y el meridiano, pasa estos datos cada

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena 	
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

segundo a un objeto de la clase **RelojModelo**, esta clase hereda de **Observable** (Observado) con el fin de que sus estados (el valor de sus propiedades o atributos) puedan ser Observados por objetos que Implementen la interfaz **Observer** (Observadores), que en nuestro caso estos observadores corresponden a objetos de la clase **RelojActionControl** ubicada en la capa de Control.

Cada vez que cambia el valor de un atributo de un objeto de tipo **Observable** “**Observado**” (**RelojModelo** y su atributo **tiempo** en nuestro ejemplo), esta clase notifica a todas sus objetos Observadores del tipo **Observer** “**Observador**” (**RelojActionControl** en nuestro ejemplo) sobre su nuevo estado (nuevo valor del atributo tiempo), estos objetos Observadores consultan el nuevo estado del objeto Observado y realizan alguna operación con esta información (en nuestro caso el objeto Observador **RelojActionControl** actualiza “**update**” el tiempo que se muestra en el objeto de tipo **RelojVista**).

A continuación veremos el código de las clases que harán parte de la capa de Modelo, esta clases RelojHilo se encarga de generar de forma concurrente información sobre el tiempo mientras va actualizando a la vista en la clase RelojVista encargada de mostrar el tiempo en una etiqueta de texto.

/**

* DatosReloj.java

*

* Clase que contine la logica para generar los datos del tiempo



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

* en formato HH:MM:SS Hora:Minutos:segundos respectivamente,

* esta clase recibe estímulos de la clase RelojControlListener

* para determinar si sigue o no generando información sobre el tiempo

*

* @autor John Carlos Arrieta Arrieta

*

*/

```
import java.util.*;
```

```
public class RelojModelo extends Observable
```

```
{
```

```
    // variable miembro cuyo valor que define el inicio
```

```
    // o parada de la información sobre el tiempo
```

```
    private boolean estado;
```

```
    private String tiempo = "00:00:00";
```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```
private RelojHilo hiloReloj = new RelojHilo();
```

```
public void setTiempo(String tiempo){
```

```
    this.tiempo = tiempo;
```

```
    setChanged();
```

```
    notifyObservers();
```

```
}
```

```
public void setEstado(boolean estado)
```

```
{
```

```
    this.estado = estado;
```

```
    if(estado == true)
```

```
    {
```

```
        iniciarReloj();
```

```
    }
```

```
}
```




Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```

public void iniciarReloj( )

{

    // CREAR UN HILO PARA QUE SE GENERAR EL TIEMPO MIENTRAS EL PROGRAMA SIGUE
    CORRIENDO

    hiloReloj.start(); // INICIAR LA EJECUCION DEL HILO

}

public String getTiempo()

{

    return tiempo;

}

public boolean getEstado()

{

    return this.estado;

```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

}

```

class RelojHilo extends Thread    {

    public void run( )

    {

        // obtengo un obejto con informacionsobre el calendario del

        GregorianCalendar calendario = new GregorianCalendar();

        int incrementoSegundo = 0;

        while( estado == true ) // while( estado )

        {

            calendario.add( Calendar.SECOND, 1);

            String M = (calendario.get(Calendar.AM_PM) == 1)? "PM" : "AM";

            String tiempoAux = calendario.get(Calendar.HOUR)+":"+

            calendario.get(Calendar.MINUTE)+":"+

            calendario.get(Calendar.SECOND)+" "+ M;

```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```
setTiempo(tiempoAux);
```

```
System.out.println("Hora: "+tiempoAux);
```

```
try
```

```
{
```

```
    Thread.sleep(1000);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
/**
```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

RelojHiloModelo.java

Clase Hilo que herada de la Clase Thread y utiliza un objeto

de la clase GregorianCalendar para genera informacion sobre el

tiempo de manera concurrenteambas.

@author John Carlos Arrieta Arrieta

*/

package sistempres.reloj.modelo;

import java.util.*;

public class RelojHiloModelo extends Thread {

// atributos

private RelojModelo relojModelo;

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```
public RelojHiloModelo() {  
  
}
```

```
public void run( )  
  
{  
  
    // obtengo un objeto con informacion sobre el calendario del  
  
    GregorianCalendar calendario = new GregorianCalendar();  
  
    int incrementoSegundo = 0;  
  
    while(relojModelo.getEstado() == true ) // iterar cada segundo  
  
    {  
  
        calendario.add( Calendar.SECOND, 1); // aumento un segundo  
  
        // obtengo el meridiano  
  
        String M = (calendario.get(Calendar.AM_PM) == 1)? "PM" : "AM";  
  
        // obtengo la Hora, los minutos y segundos  
  
        String tiempoAux = calendario.get(Calendar.HOUR)+":"+  
  
        calendario.get(Calendar.MINUTE)+"+";
```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```
calendario.get(Calendar.SECOND)+" "+ M;
```

```
// cambio el estado del tiempo en la clase observada
```

```
relojModelo.setTiempo(tiempoAux);
```

```
try
```

```
{
```

```
    Thread.sleep(1000); // suspendo al hilo por un segundo
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
}
```

```
}
```

```
}
```

```
public void setRelojModelo(RelojModelo relojModelo){
```

```
    this.relojModelo = relojModelo;
```

```
}
```

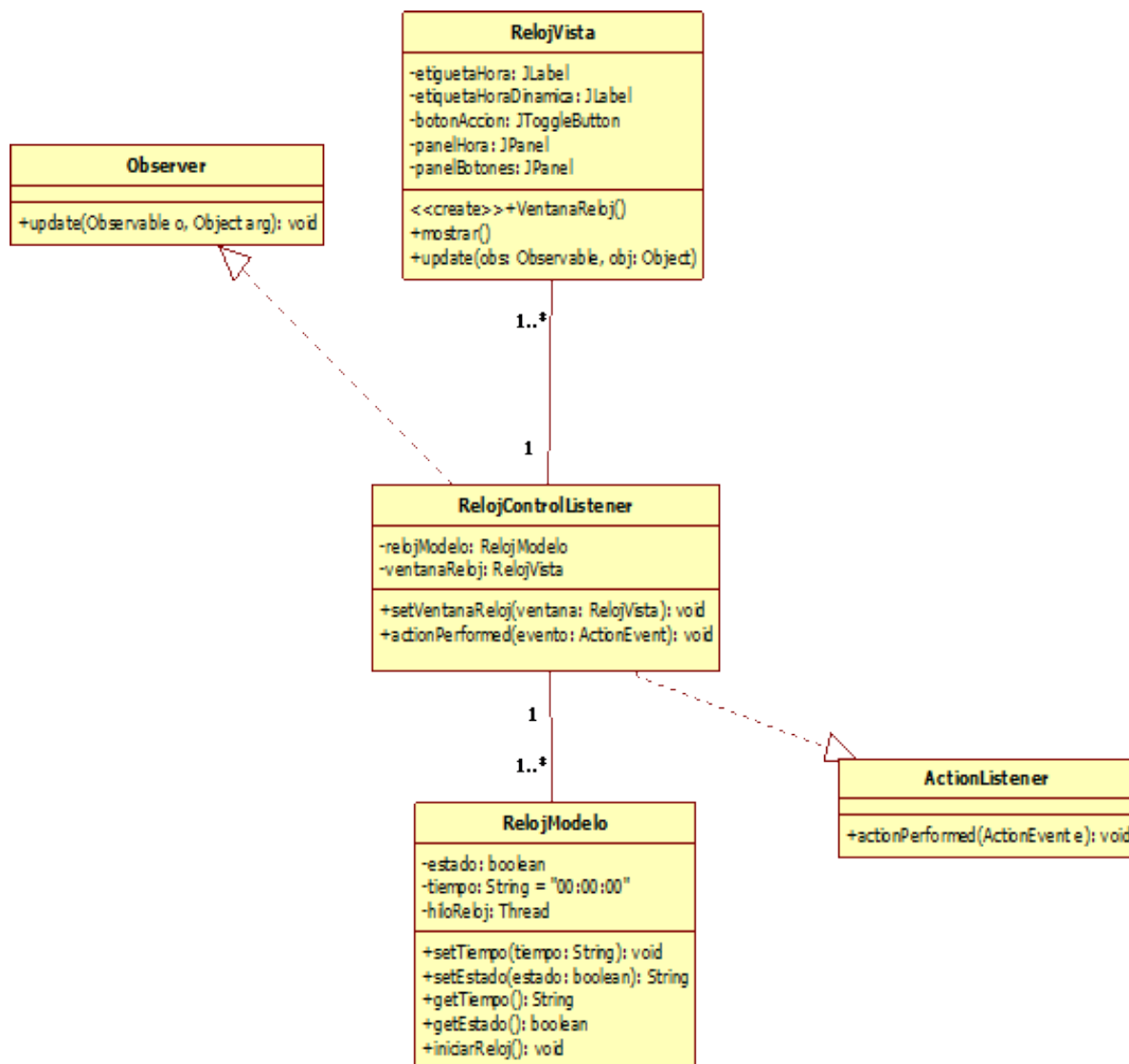



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

}

La clase de la capa de Control

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

En la capa de control he colocado una clase llamada **RelojActionControl**, cuyo código es ejecutado automáticamente cada vez que se produce un evento de acción (**ActionEvent**) sobre el botón de inicio o parada que se encuentra en la clase **RelojVista** (la vista del programa), esta ejecución automática se debe gracias a que esta clase implementa la interfaz **ActionListener**, esta interfaz obliga a las clases que la implementen (**RelojActionControl**) a definir el código necesario que deben ejecutar automáticamente cada vez que ocurra un evento del tipo **ActionEvent**, este código se debe escribir en el método (función) llamado **actionPerformed(ActionEvent evento)**, el cual recibe como parámetro un objeto de tipo **ActionEvent**, que contiene toda la información necesaria sobre el evento, las clases que implementen la Interfaz ActionListener se convierten en Oyentes (Listeners) de del evento **ActionEvent**

Esta clase **RelojActionControl** a su vez, implementa de igual forma a la interfaz **Observer**, convirtiéndose automáticamente en una clase Observadora, cuyo interés se centra en el cambio del **atributo tiempo** propio de la clase Observada **RelojModelo**.

Esta clase actúa como controlador del software porque:

- Su responsabilidad es registrarse como observador del modelo.
- Actualiza en la capa de Vistas (la ventana **RelojVista**), los datos que que provienen de las operaciones se ejecutan en la capa de Modelo (**RelojModelo**).
- Re-direcciona los eventos que provienen de la capa de Vistas (operación Parar y Detener del botón de la ventana **RelojVista**) hacia las correspondientes operaciones del modelo (**RelojModelo**) que ejecutan dichas estas peticiones del usuario.



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

A continuación coloco el código fuente de la clase **RelojActionControl**

```
/**
 * RelojControlListener.java
 *
 * Clase que escucha los eventos ocurridos sobre el boton de accion
 * verifica el estado del mismo, si esta pulsado, entonces inicia
 * el reloj, si no está seleccionado, entonces para el reloj
 *
 * @autor John Carlos Arrieta Arrieta
 *
 */

package sistempres.reloj.control;

import java.awt.event.*;
```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```
import java.awt.*;
```

```
import javax.swing.JToggleButton;
```

```
import sistempres.reloj.modelo.RelojModelo;
```

```
import sistempres.reloj.vistas.RelojVista;
```

```
import java.util.Observer;
```

```
import java.util.Observable;
```

```
public class RelojActionControl implements ActionListener, Observer{
```

```
    private RelojModelo relojModelo;
```

```
    private RelojVista relojVista;
```

```
    public void setReloVistaj(RelojVista vista)
```

```
    {
```

```
        relojVista = vista;
```

```
    }
```

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```

public void actionPerformed(ActionEvent evento)

{

    if(evento.getSource() instanceof JToggleButton)

    {

        JToggleButton botonAccion = (JToggleButton)evento.getSource();

        if(botonAccion.isSelected()){

            if(relojModelo == null){

                relojModelo = new RelojModelo();

                relojModelo.addObserver(this);

            }

        }

        System.out.println("Seleccionado "+botonAccion.isSelected());

        boolean estado = botonAccion.isSelected();

        if(estado == true)

        {

            botonAccion.setText("Parrar");

```




Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

}

else

{

botonAccion.setText("Iniciar");

}

relojModelo.setEstado(estado);

}

}

public void update(Observable obs,Object obj)

{

if(obs instanceof RelojModelo){

relojVista.getEtiquetaHora().setText(relojModelo.getTiempo());

}

}

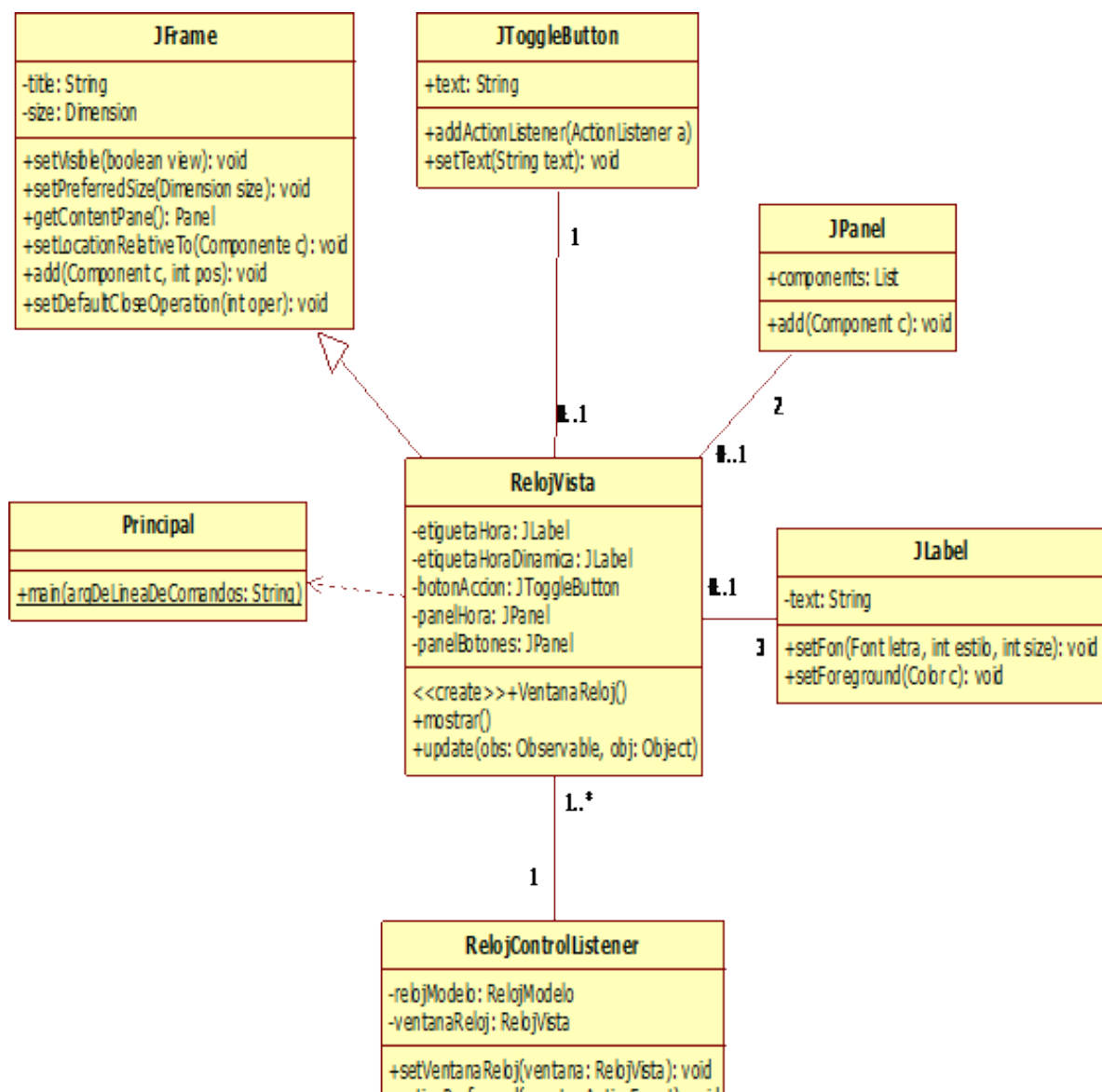
}



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Las clases de la capa de Vistas

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	






Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Esta capa es la encargada de presentar al usuario la información del tiempo, producida en la capa de Modelo, está compuesta básicamente por un conjunto de clases, cuya finalidad es crear objetos que de interfaz grafica de usuario (GUI), entre los que se encuentran:

- Un objeto ventana creado con la clase **RelojVista** que hereda a su vez de la clase **JFrame** del paquete **javax.swing** del JSDK. Su referencia en el código es una variable llamada **vista**, declarada en el código de la clase **Principal**
- Un objeto etiqueta creado con la clase **JLabel** del paquete **javax.swing** del JSDK, este objeto se utiliza para mostrar de forma estática mi Nombre en la ventana **RelojVista**. Su referencia en el código es una variable llamada **etiquetaAutor**, declarada en el código de la clase **RelojVista**.
- Un objeto etiqueta creado con la clase **JLabel** del paquete **javax.swing** del JSDK, este objeto se utiliza para mostrar el mensaje estático “**La hora es:**” en la ventana **RelojVista**. Su referencia en el código es una variable llamada **etiquetaMsg**, declarada en el código de la clase **RelojVista**.
- Un objeto etiqueta creado con la clase **JLabel** del paquete **javax.swing** del JSDK, este objeto se utiliza para mostrar de forma dinámica información sobre el tiempo en formato “**hh:mm:ss M**” en la ventana **RelojVista**, esta información es suministrada por el objeto de tipo **RelojActionControl** de la capa de Control, objeto que a su vez obtiene dicha información actualizada gracias a las notificaciones que recibe del objeto **RelojModelo** de la capa de Modelo. Su referencia en el código es una variable llamada **etiquetaHora**, declarada en el código de la clase **RelojVista**.
- Un objeto panel creado con la clase **JPanel** del paquete **javax.swing** del JSDK, este objeto se utiliza para colocar la **etiquetasMsg** y la **etiquetaHora**, este panel a su vez es colocado en el centro de la ventana **RelojVista**, Su referencia en el código es una variable llamada **panelHora**,

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena		PIENSA EN GRANDE Piensa en UNISINÚ
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas	
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27	
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI	
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 		

declarada en el código de la clase **RelojVista**.

- Un objeto botón-alternado creado con la clase **JToggleButton** del paquete **javax.swing** del JSDK, este objeto se utiliza para solicitar las operaciones de Inicio y Parada del reloj digital desde la ventana **RelojVista**. Su referencia en el código es una variable llamada **botonAccion**, declarada en el código de la clase **RelojVista**.
- Un objeto panel creado con la clase **JPanel** del paquete **javax.swing** del JSDK, este objeto se utiliza para colocar el **botónAccion** en la ventana, este panel a su vez es colocado en la parte inferior (SOUTH) de la ventana **RelojVista**, Su referencia en el código es una variable llamada **panelBotones**, declarada en el código de la clase **RelojVista**.

A continuación coloco el código de la clase **RelojVista**

```
/**
```

```
RelojVista.java
```

```
Clase diseñada para mostrar la informacion del tiempo generada por la capa
```

```
de modelo, actualizada por la capa de control
```

```
@author John Carlos Arrieta Arrieta
```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

*/

```
package sistempres.reloj.vistas;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
import java.awt.*;
```

```
import java.util.*;
```

```
import sistempres.reloj.control.RelojActionControl;
```

```
public class RelojVista extends JFrame
```

```
{
```

```
    // variables miembro o atributos del objeto VentanaReloj
```

```
    JLabel etiquetaHoraMsg;
```

```
    JLabel etiquetaHora;
```

```
    JLabel etiquetaAutor;
```

```
    JToggleButton botonAccion;
```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

JPanel panelHora;

JPanel panelBotones;

RelojActionControl control = new RelojActionControl();

/*

constructor de la clase para podr Craer un objeto de

tipo VentanaReloj con todos sus otros objetos.

*/

public RelojVista()

{

super();

setTitle("Reloj Digital MVC con Java y el IDE GEL");

etiquetaAutor = new JLabel("John Carlos Arrieta Arrieta");

etiquetaAutor.setHorizontalAlignment(JLabel.CENTER);

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```
etiquetaAutor.setFont(new Font("Arial",1,20));
```

```
etiquetaAutor.setForeground(Color.YELLOW);
```

```
etiquetaHoraMsg = new JLabel("La hora es:");
```

```
etiquetaHoraMsg.setFont(new Font("Arial",1,18));
```

```
etiquetaHoraMsg.setForeground(Color.BLUE);
```

```
etiquetaHora = new JLabel("00:00:00");
```

```
etiquetaHora.setFont(new Font("Courier New",1,18));
```

```
etiquetaHora.setForeground(Color.RED);
```

```
botonAccion = new JToggleButton("Iniciar");
```

```
botonAccion.setMnemonic('I');
```

```
panelHora = new JPanel();
```

```
panelHora.add(etiquetaHoraMsg);
```

```
panelHora.add(etiquetaHora);
```

```
panelBotones = new JPanel();
```

```
panelBotones.add(botonAccion);
```

```
add(etiquetaAutor, BorderLayout.NORTH);
```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```

add(panelHora, BorderLayout.CENTER);

add(panelBotones , BorderLayout.SOUTH);

control.setReloVistaj(this);

botonAccion.addActionListener(control);

setDefaultCloseOperation(EXIT_ON_CLOSE);

}

// funcion o metodo de la clase VentanaReloj que permite

// mostrar la ventan en la Pantalla del PC

public void mostrar()

{

    pack();

    setLocationRelativeTo(null);

    setVisible(true);

}

public JLabel getEtiquetaHora(){

```



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

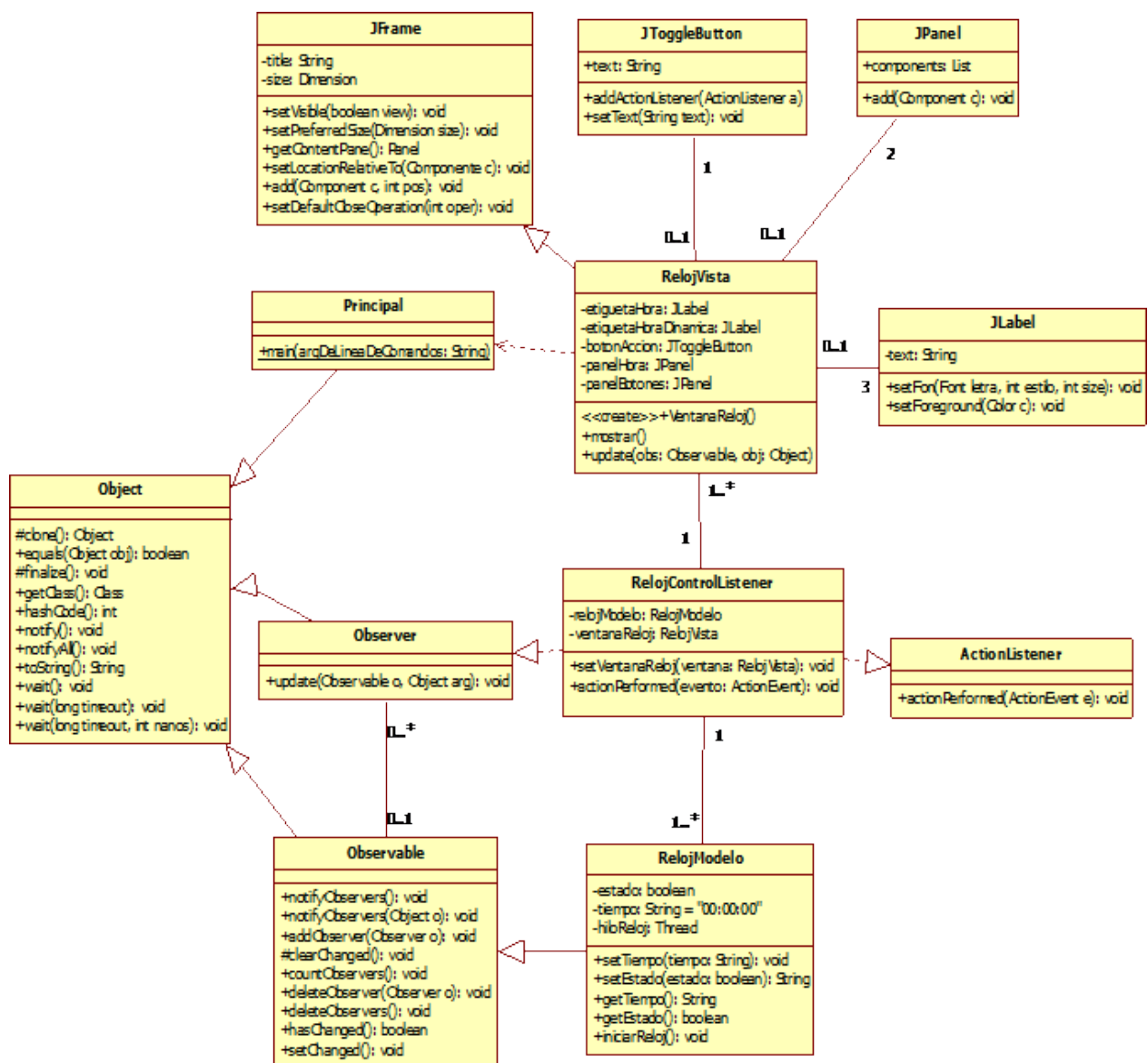
```
return etiquetaHora;
```

```
}
```

```
}
```

La gráfica siguiente representa el Diagrama de Clases completo del software que diseñe para este artículo.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entorno Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

ejecución



Saludos amigos...

Continuando con la saga de artículos sobre Recordando el Pasado y Agradeciendo el Presente, esta vez les traigo a colación una herramienta que nació hace ya una década y que resulta ser poco conocida por la mayoría de docente y estudiantes de Java, se trata de subir un nivel mas que el desarrollo con bloc de notas, la herramienta que mostrare se llama GEL, se trata de un IDE (Entorno Integrado de Desarrollo) construido por una comunidad de desarrolladores libres, GEL inicialmente fue desarrollado para ser un IDE java para GNU/Linux, pero gracias a su gran desempeño y funcionalidad para desarrollo de aplicación Java, rápidamente fue tomando adeptos y sus desarrolladores publicaron una versión compatible con sistemas operativos Windows.

Muchas instituciones de educación que enseñan a programar software con Java utilizan software



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

en propietario , en especialmente un producto gratis (de libre descarga e instalación, pero no podemos tener acceso a su código fuente), llamado JCreator LE, esta versión al ser gratis tiene muchas restricciones de funcionalidad deshabilitadas, solo podemos tener acceso a las funcionalidades deshabilitadas si compramos legalmente un copia de su licencia comercial llamada JCreator PRO, algo que las universidades no hacen, algunas solo se limitan a utilizar la versión LE y otras simplemente cometen actos ilícitos permitiendo que en sus equipos de compute se utilice una copia Pirata (software privativo utilizado con una copia adquirida ilegalmente, violando las condiciones que estipula su original propietario en las clausulas de su licencia). El problema no es utilizar JCreator o cualquier otro software para realizar alguna tareas especifica, de hecho JCreator en su versión Pro (Profesional) es muy buen producto, el problema es utilizarlo de forma ilegal (Pirata), más aun, cuando este tipo de prácticas anti-valores y deshonestas son heredadas por sus propios estudiantes, cultivando desde muy temprano conductas anti éticas a lo largo de su carrera profesional.

Uno de los tantos beneficios que tiene el uso de software libre, es precisamente carecer de la necesidad de utilizar (piratear) software de forma ilegal, o por el contrario, realizar altas inversiones con el fin de adquirir legalmente costosas licencias de software privativo.

Hoy por hoy, hay no solo una sino varias alternativas de software libre para los software privativos más populares del mercado, si les queda alguna duda de lo que estoy afirmando, realicen ustedes mismo una pequeña búsqueda por la Web de productos privativos, sistemas operativos, productos de multimedia, productos educativos etc., y encontraran una larga lista de opciones con licencias de software libre.

El caso de GEL resulta ser muy curioso ya que se desarrollo primero que su contrapartida privativa JCreator y resulta que son muy parecidos en cuanto a funcionalidad y paraciencia, con la diferencia que GEL pose muchas más funcionalidades que JCreator LE, incluso más que la



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

versión comercial PRO, y no lo voy a decir solo por decirlo, en este articulo hare algunas comparaciones entre ambos IDE.

Estos dos IDE tuvieron mucha aceptación por la comunidad académica y desarrolladores Java, hace una década gozaban de mucha gloria, aun cuando ya para entonces se comenzaban a consumarse y a tomar fuerza los IDE que hoy por hoy conocemos como IDEs de IV generación, tales como NetBeans, JBuilder, Eclipse, Visual J++, Sun One Studio y muchos más.

Descargas

GEL <http://www.gexperts.com> (tienes que hacer parte de su comunidad registrándote a un grupo de yahoo)

JCreator <http://www.jcreator.com/downloadesarad.htm>

Ventajas de GEL IDE for Java:

1. Es software libre
2. Su código fuente disponible, mejoras según nos convenga, costos de adquisición e instalación nulos
3. Es un IDE estable con muchas funcionalidad para desarrollar aplicaciones en java
4. Tiene amplia documentación
5. Es muy rápido y estable



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Desventajas de GEL IDE for Java

1. El proyecto ya no tiene comodidad de desarrollo desde 2003
2. Al disolverse su comunidad, el producto carece de soporte, de documentación actualizada, de actualizaciones y respaldo

Ventajas de JCreator:

1. Es un producto Gratis en constante evolución
2. Su soporte y documentación seguros
3. Es un IDE confiable con muchas funcionalidad para desarrollar aplicaciones en java
4. Es rápido y estable

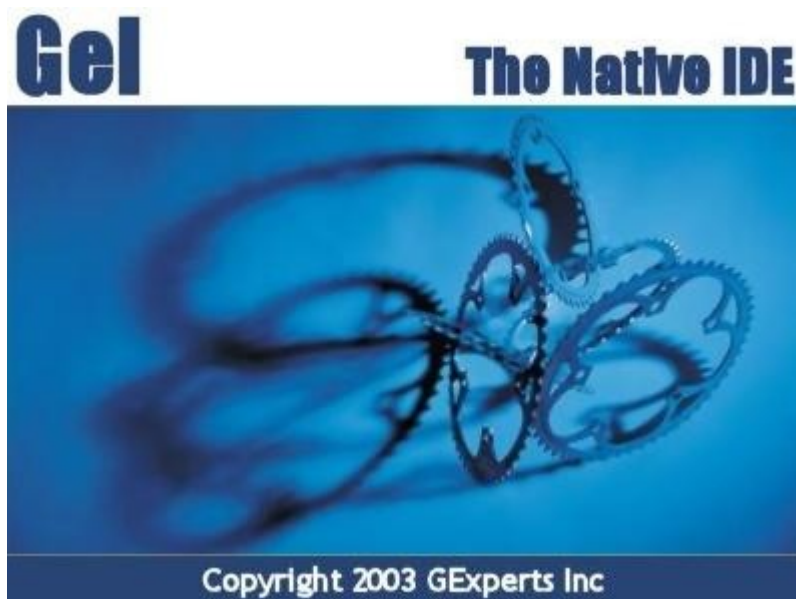
Desventajas de JCreator:


1. Es un producto gratis pero No Libre, no disponible su código fuente
2. Su actualización y mejoras dependen del proveedor
3. La documentación es suministrada por su proveedor
4. Solo la versión Comercial no gratis tiene todas sus funcionalidades activas
5. Su estabilidad y rendimiento no ha sido probada en Linux y generalmente se utiliza en este S.O con Windows de forma virtualizada

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

La instalación de ambos productos es típica de cualquier instalación de software para plataforma Windows, presentan un asistente común , el cual solicita un dialogo al cual solo hay que responderle las preguntas o dejar las opciones que tiene por defecto.

Iniciando el desarrollo utilizando GEL



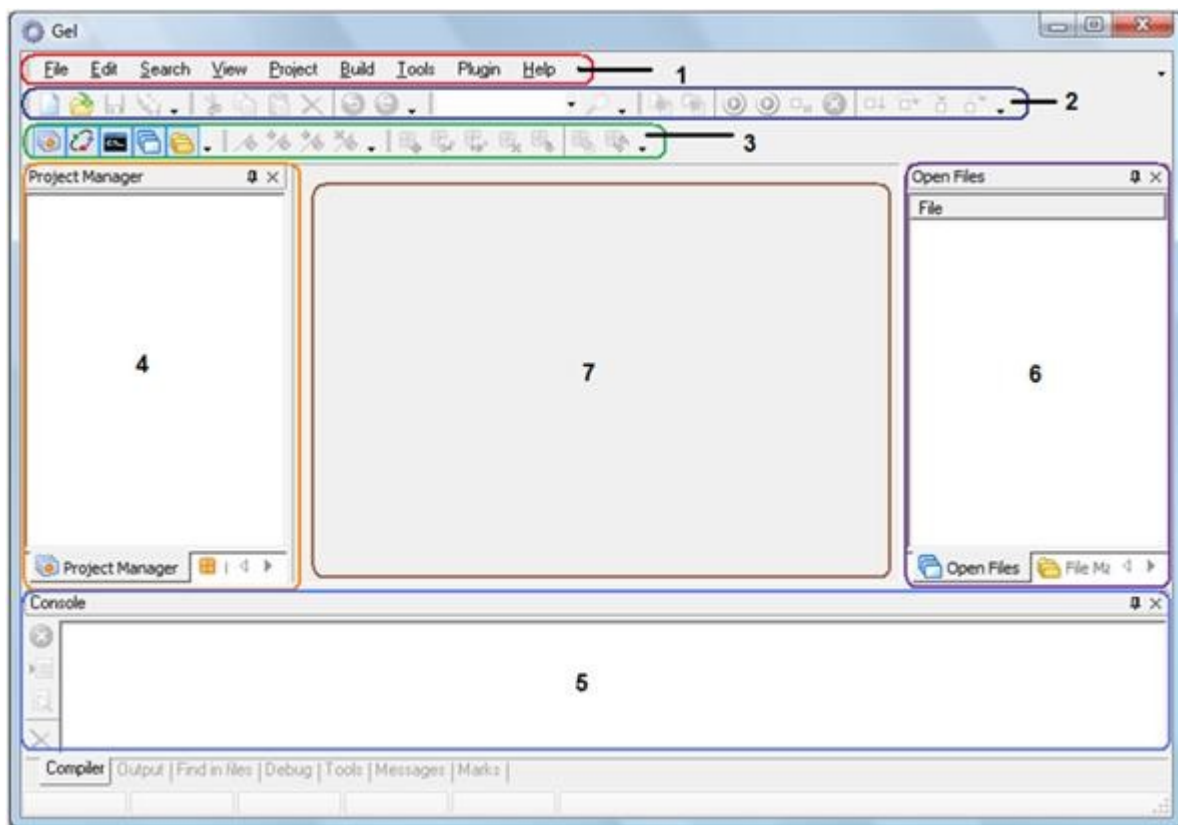
 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena		PIENSA EN GRANDE Piensa en UNISINÚ
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas	
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27	
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI	
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 		



Splash informativo al inicio del programa y ventana de Aviso del día

GEL es un IDE bajo licencia libre GPL para desarrollo de aplicaciones con el Lenguaje Java, es la mayor y más fuerte alternativa para el IDE privativo JCreator, son muy similares en apariencia y nivel de funcionalidad.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

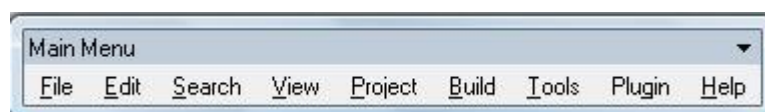


Ventana

Principal de GEL

He enumerado y rodeado con un color diferente a las partes más importantes de la ventana, con el fin de de hacer una breve explicación de cada una de ellas.

1. Barra de menú principal,



en ella se

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

encuentran todas las opciones funcionales que ofrece el programa accesible mediante comandos de Menú e ítems de Menú, los cuales ejecutan acciones inmediatas o muestran una ventana de dialogo, desde la que se pueden configurar las acciones deseadas.

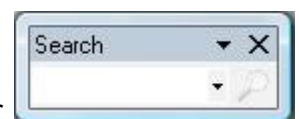
2. Son barras de herramientas con los comandos más importantes del menú Archivo



, el menú Edición



, el menú Buscar



y el

menú Construir



3. Son barras de herramientas con los comandos más importantes del menú Ver



el submenú marcas del menú Ver



y el submenú Control de Fuentes del menú

Herramientas (tools)



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



4. Es un panel lateral desplegable llamado Manejador de Proyectos

Este panel es uno de los elementos más utilizados durante el desarrollo de un proyecto con Java utilizando GEL como IDE, el Manejador de Proyectos contiene un árbol con cuyos nodos corresponden a los ficheros o recursos (código fuente, imágenes, librerías, archivos de configuración XML, entro otros) que contiene un Proyecto.

5. Es un panel inferior desplegable llamado panel de Consola,

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



el cual contiene

subpaneles que muestran información sobre aspectos importantes del proyecto, aspectos relacionados con la información de compilación y ejecución del proyecto, búsquedas de palabras o recursos que están o pueden estar en el proyecto,

Mensajes del IDE y del la plataforma SO y las marcas que hacemos en el código fuente.

6. El otro panel lateral desplegable llamado Archivo Abierto,

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Contiene un árbol cuyos nodos representan información sobre los elementos que estructuran el contenido del archivo abierto actualmente, este archivo abierto puede ser el código fuente de una clase, una enumeración o una Interfaz Java.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	





7. Es el panel de Edición de Código Fuente

Es quizás el elemento más utilizado dentro de un IDE, puesto que dentro de este panel GEL coloca el contenido de los archivos de código fuente .java, con el fin de que podemos editarlo (agregarle o quitarle elementos de código fuente java, como clases, interfaces, enumeraciones, propiedades, métodos, bloques, etc.)

El editor de código fuente de GEL resalta en colores varios las palabras claves del lenguaje Java, esto permite al programador, visualizar e identificar con mayor detalle los elementos del lenguaje, minimizando las posibilidades de confusión visual que genera la lectura del código fuente de un programa.

Manos a la Obra

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena 	
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Para explicar la utilización de este IDE como herramienta gestora de un proyecto de software con el lenguaje de programación Java, voy a construir un pequeño programa que actúa como reloj digital, el cual presenta información del tiempo en formato HH:MM.SS M, dicho reloj podrá ser iniciado y parado mediante un botón.

Durante el desarrollo de este proyecto utilizare conceptos de programación útil e igualmente importante, entre los que podemos mencionar Hilos (Threads) que sirven para realizar operaciones concurrentes (se ejecutan al mismo tiempo, mientras que se ejecuta el programa principal) y MVC como patrón de diseño arquitectónico para la construcción de software.

Además del uso de Interfaces, herencia, clases anidadas, trabajo con ventanas, etiquetas de texto, botones y eventos.

Creando un proyecto en GEL

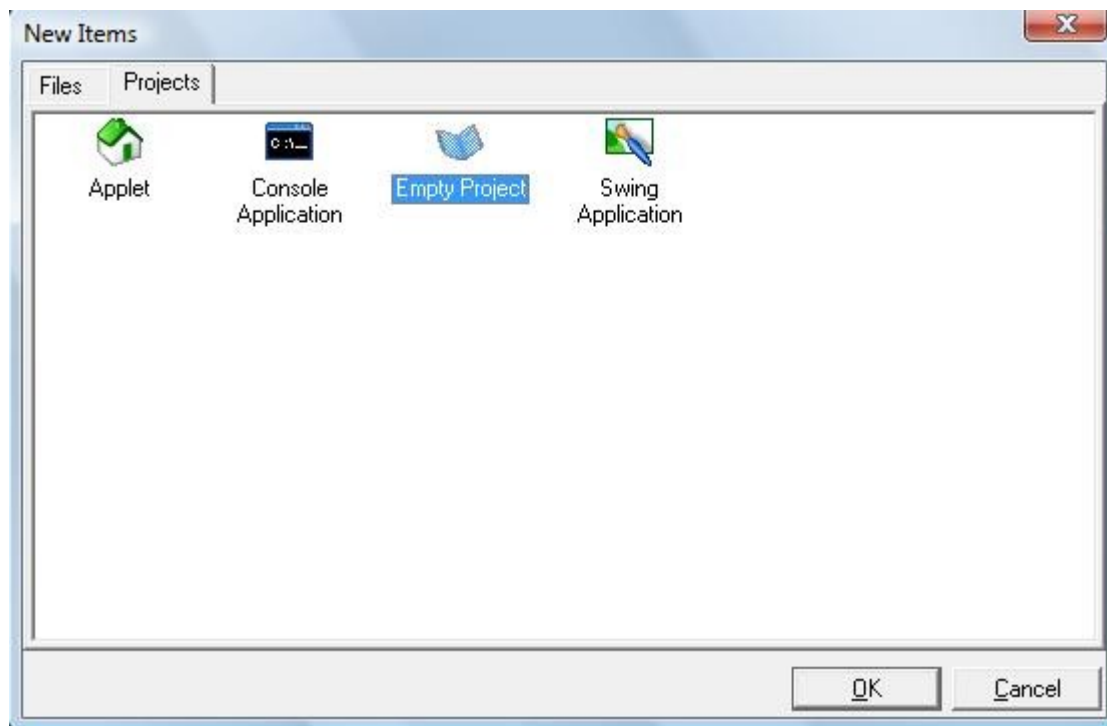
GEL se puede utilizar para crear, explorar o editar un archivo de código fuente .java de forma aislada (no vinculado a ningún proyecto), pero también se puede utilizar para gestionar todos los archivos o recursos que hacen parte de un proyecto de software, recursos que se encuentran relacionados entre si y que dan forma o estructura al programa.

Para poder trabajar de forma mas cómoda y organizada creare un Proyecto para manejar de forma segura todos los recurso de Reloj Digital.

Un poco de notación que utilizare para avrevier la descripción de pasos en el IDE GEL, de ahora en adelante >> significa seleccionar la próxima opción, Clic es hacer un solo clic Izquierdo, 2Clic es doble clic Izquierdo, clicD es clic derecho,

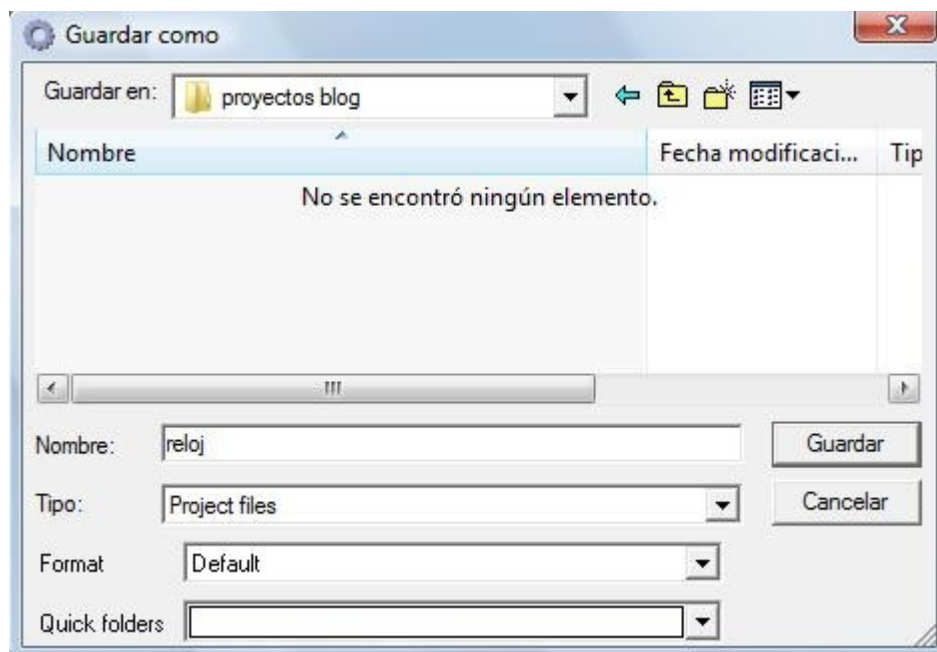
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Clic en Menu Archivo >>Nuevo



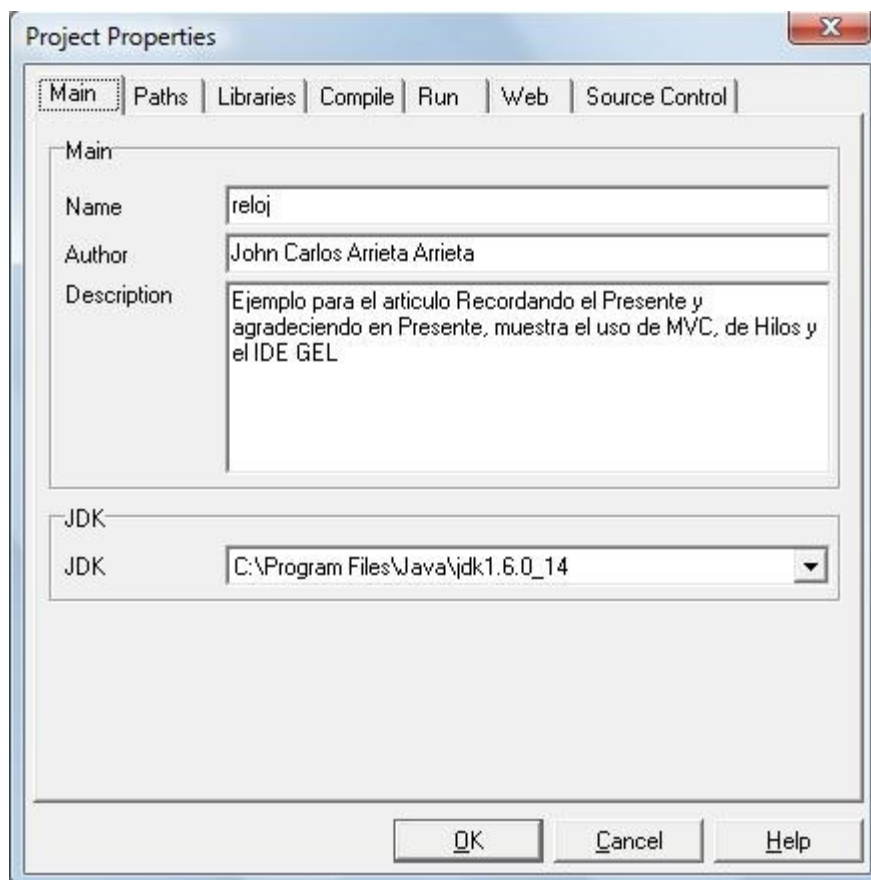
Clic en la pestaña Proyectos, 2Clic Proyecto Vacio.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Dar un nombre al proyecto (se recomienda que no tengan espacios en blanco y solo combinación de número y/o letras, no iniciar con número y todo en minúscula), en nuestro caso lo llamaremos reloj

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

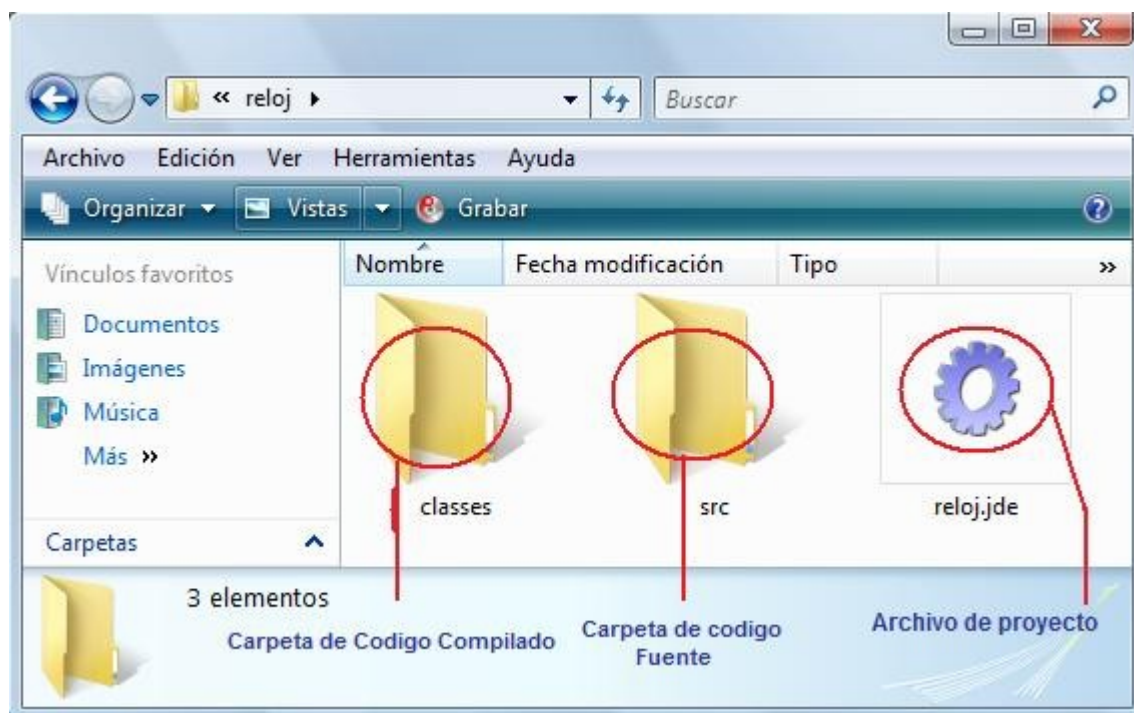
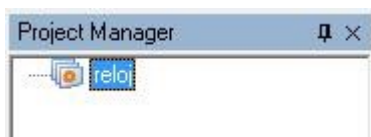


Configurar las propiedades del nuevo proyecto, esto permite definir el autor, la ruta de la variable classpath para el código fuente y el código compilado, las librerías que deseamos agregar al proyecto, aspectos de compilación y ejecución, aspectos del entorno Web si el proyecto es de tipo Web y el control de código fuente.

Al crearse el proyecto GEL agrega un nodo raíz en el árbol de panel de manejador de proyectos,

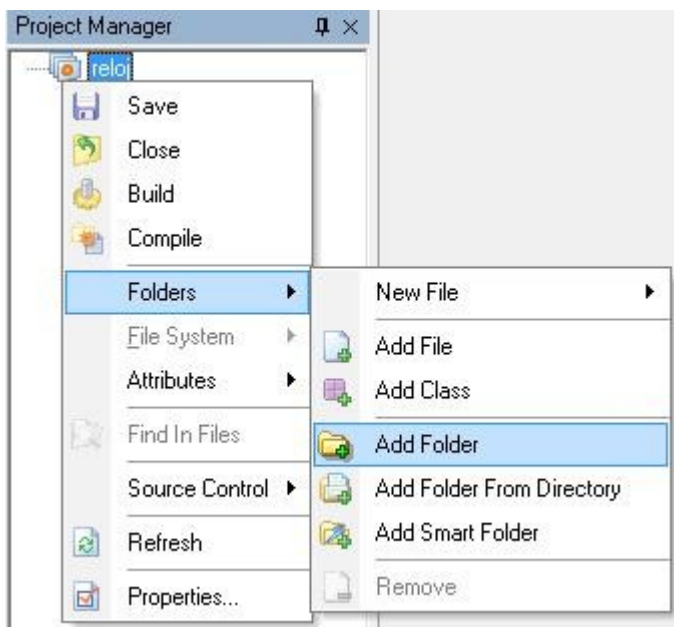
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

este nota o ITEM tiene el mismo nombre del proyecto, en realidad este ítem representa el archivo de configuración del proyecto, el cual se encuentra en la carpeta del proyecto que es igualmente creado por GEL en la ruta que le indicamos.



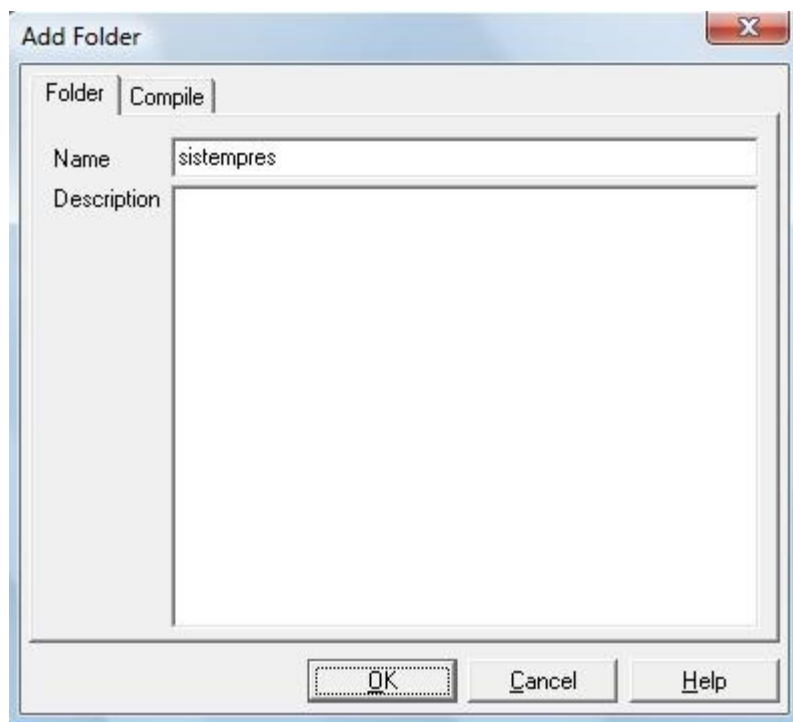
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Ahora procedemos a agregar los elementos o recursos del proyecto, para ello hacemos clicD sobre el Item del proyecto en el Panel Manejador de Proyectos, luego en el menú contextual que aparece



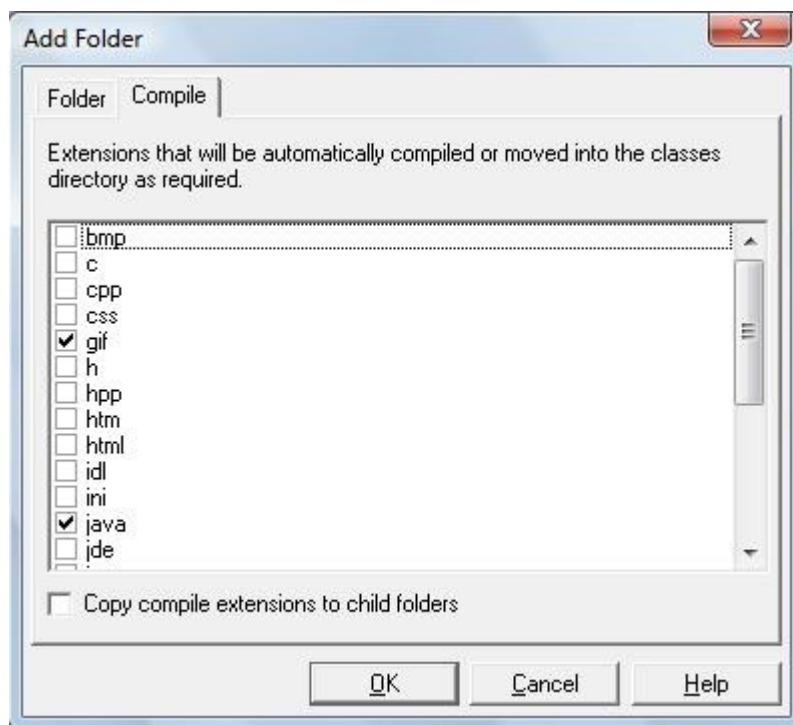
Seleccionamos Agregar Carpeta (un paquete del proyecto)

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

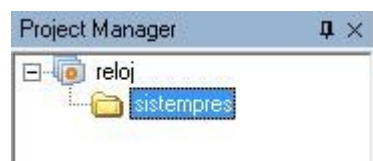


De colocamos un nombre (no puede tener espacios, puede tener solo combinaciones entre números y letras o carácter guion de bajo _, no puede iniciar en número), la descripción del paquete es opcional

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Podemos escoger que tipo de archivo será compilado y colocados en este paquete una vez se realice el proceso de compilación, por defecto aparecen ficheros .gif (imagen) y ficheros .java (código fuente Java), podemos ver como aparece ente nuevo recurso en el árbol del panel



Manejador del Proyectos, además podemos observar que este nuevo recurso pertenece al proyecto reloj que acabamos de crear.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Los paquetes se utilizan para organizar los recursos (ficheros) del proyecto según su funcionalidad u objetivo de los mismos en el proyecto, en mi caso voy a crear un paquete principal o raíz, del cual dependerán el resto de paquetes del proyecto reloj, para hacerlo simplemente me nos colocamos sobre el ítem del paquete raíz (sistempres) del manejador de Proyecto, damos clicD y repetimos los pasos para crear un nuevo paquete llamado reloj, dentro de este paquete creamos tres paquetes hijos llamados modelo, control y vistas respectivamente, la estructura de paquetes debería quedar como se muestra en la siguiente figura



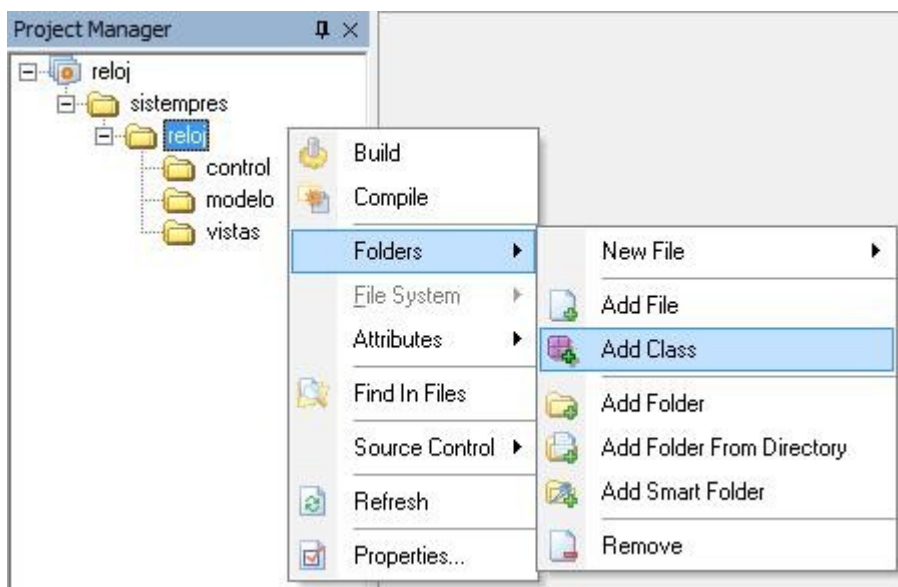
Para escribir el programa utilizaremos un patrón arquitectónico llamado MVC explicado en la [II parte](#) de este artículo

Agregando Clases al Proyecto

Agregando la clase Principal al proyecto, por donde iniciar nuestro programa Reloj, para hacerlo damos clicD sobre el paquete reloj que se encuentra en la jerarquía de paquetes del Manejador de Proyectos, luego

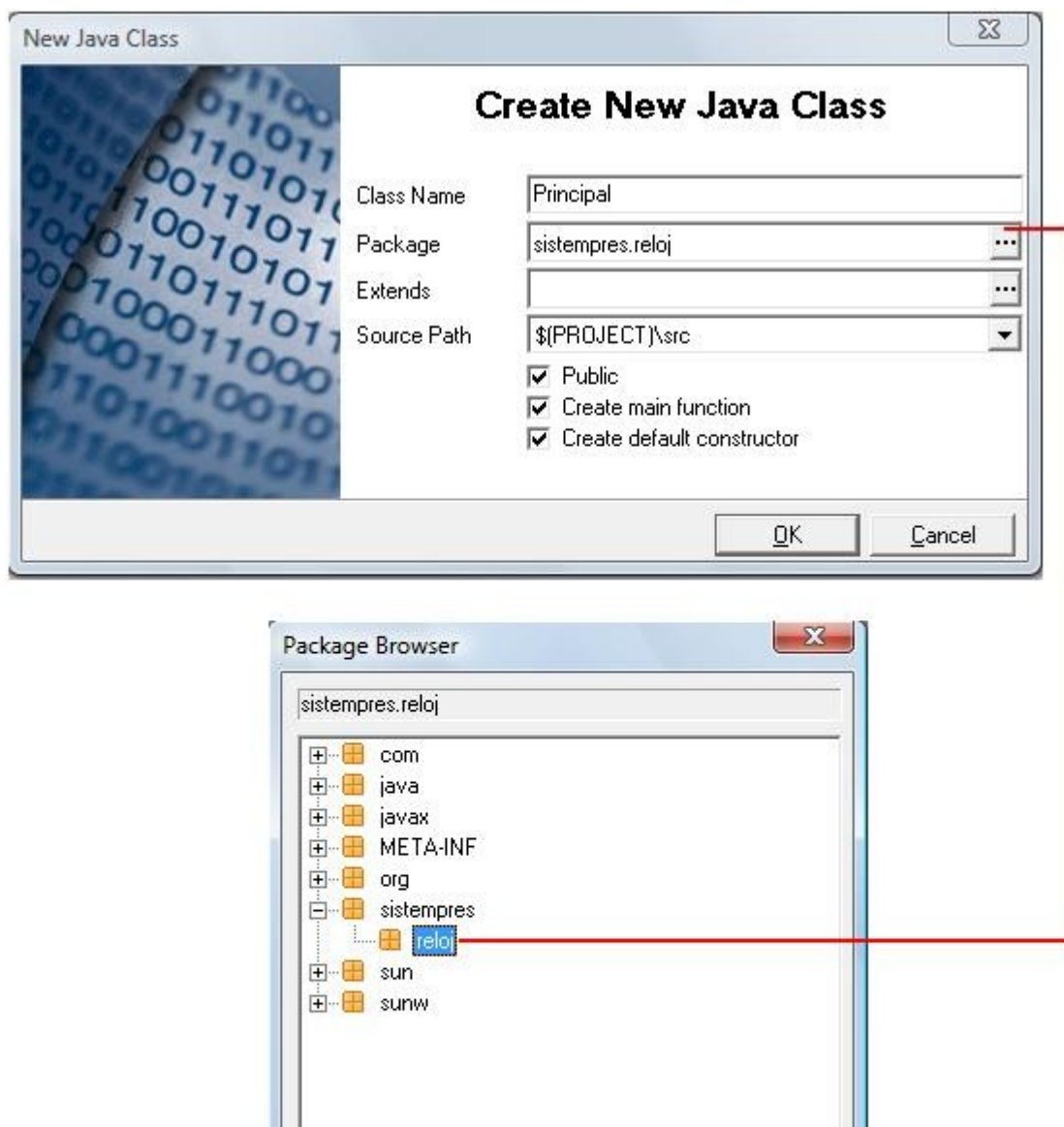
```
>> Carpeta >> Agregar Clase
```

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Nos aparece la ventana para crear una clase Java, aquí configuramos el nombre de la clase que en este caso se llamara **Principal.java** y hará parte del paquete **sistempres.reloj**, también se le indica a Gel que la clase debe ser publica, que debe agregar una constructor por defecto y además que debe tener el método public static void main(String arg[]), para que el programa pueda iniciar su ejecución por aquí.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

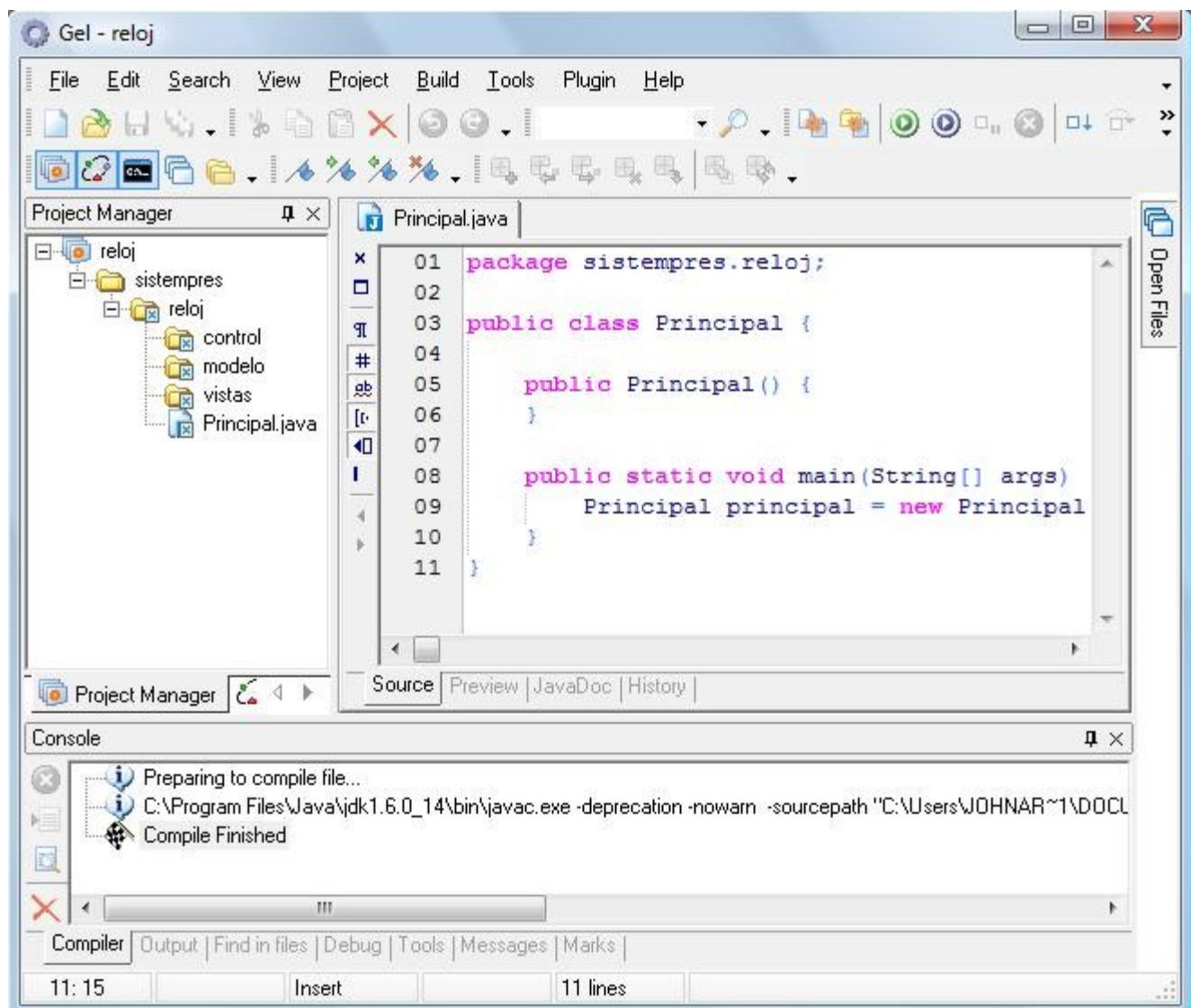





Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Al pulsar clic en OK, se puede ver como GEL genera un fichero llamado Principal.java, lo coloca dentro del paquete indicado y muestra el código fuente generado dentro del editor, además podemos ver como el editor aplica colores a las palabras clave del lenguaje Java, para diferenciarlas del resto de palabras del código fuente.

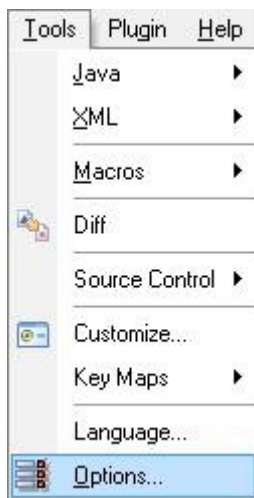
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena		PIENSA EN GRANDE Piensa en UNISINÚ
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas	
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27	
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI	
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 		

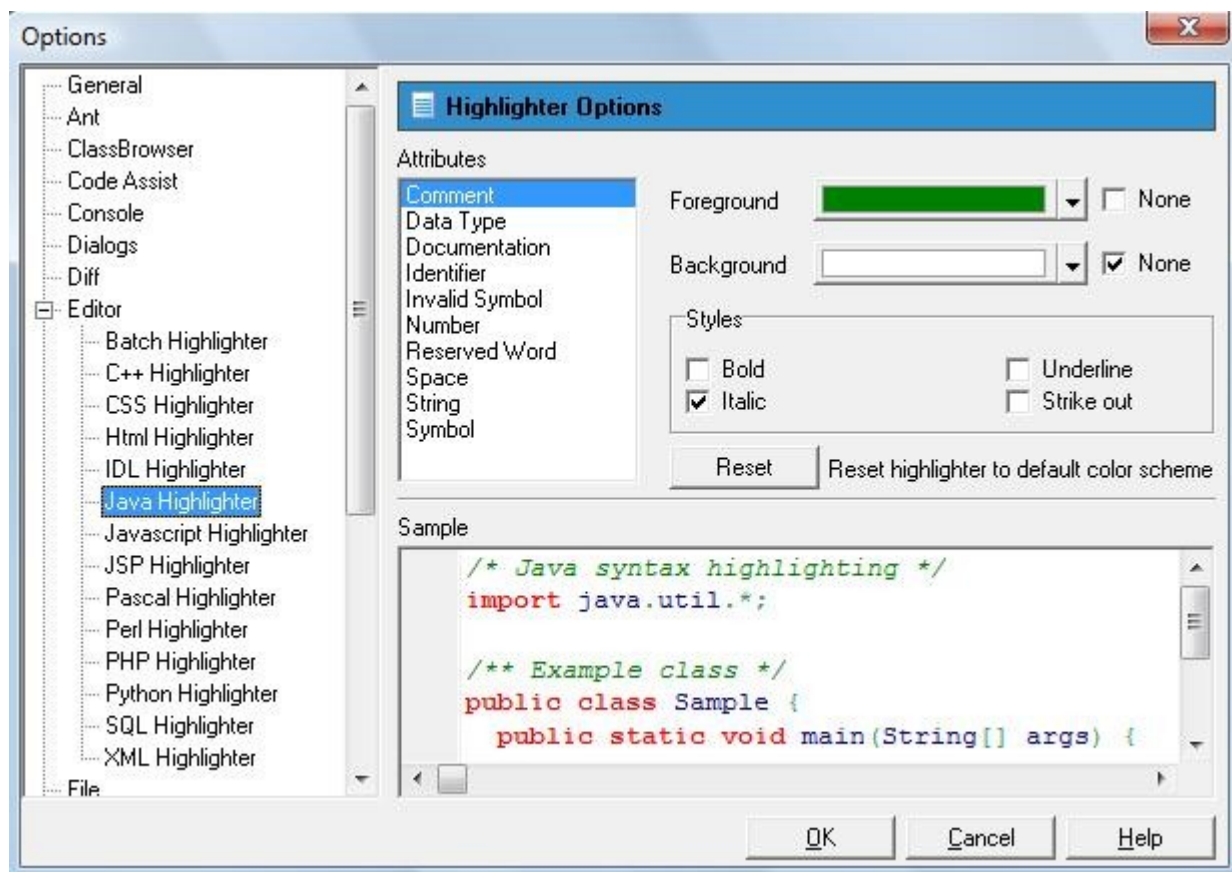
Cambiar las características del editor de código fuente

Se puede cambiar las características y comportamiento del editor de código fuente en el ítem Opciones del Menú Herramientas



El cual despliega la siguiente venta de dialogo

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Luego en el árbol lateral izquierdo hay que seleccionar el Item **Java Highlighter**, seleccionamos un elemento del código fuente en la lista de atributos, entonces se le puede cambiar el color de fuente en Foreground y el color de fondo en Background, además podemos aplicar estilos como Negrita, Cursiva, Subrayado y tachado, en el panel ejemplo tenemos una vista previa de cómo quedarían el código después de aplicar los cambios deseados.

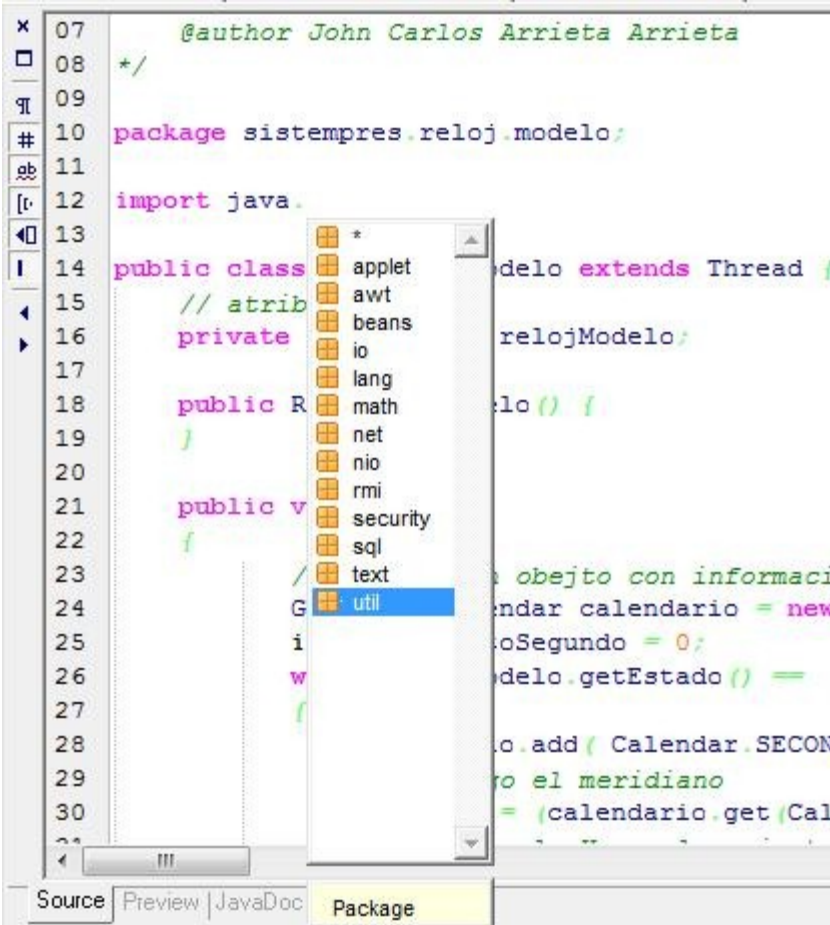


Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Completado automático de código

GEL al igual que muchos otros IDE posee la capacidad de autocompletar el código al mismo tiempo que vamos escribiendo una palabra clave del lenguaje Java, para activar el autocompletado de código tenemos que presionar la siguiente combinación de teclas SHIFT + CTRL + ESPACIO, alternatively también puedes utilizar SHIFT + CTRL + W, esta acción muestra una lista de clases y/o paquetes emergentes disponibles tanto en el JSDK como en los repositorios adicionales que estamos utilizando.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

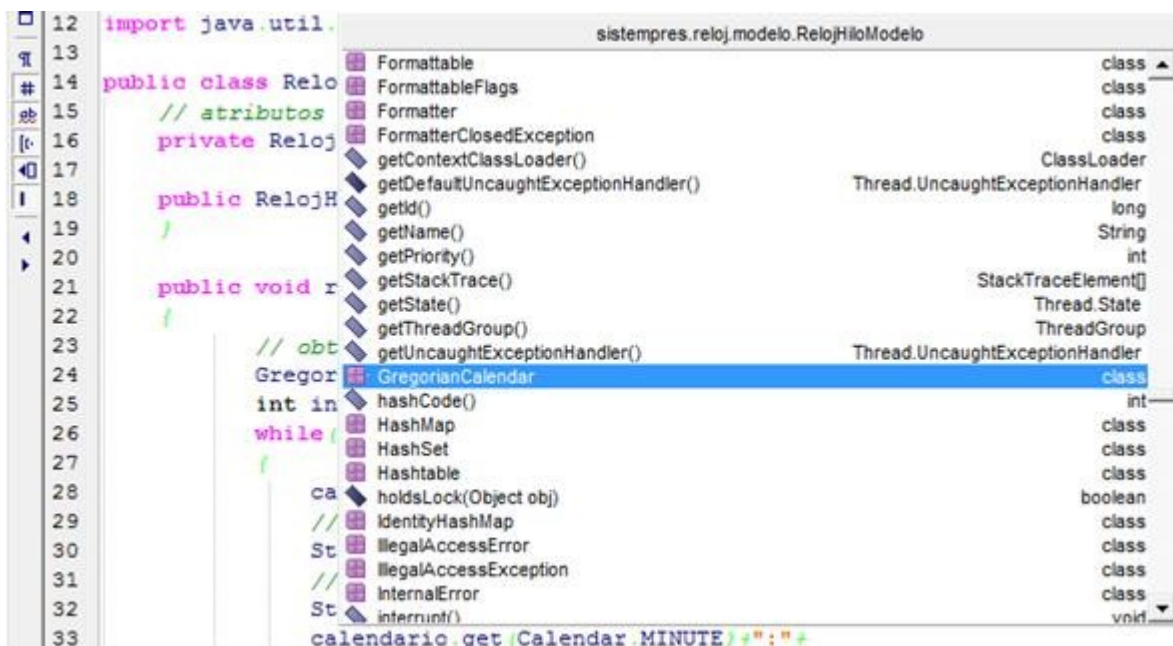


```

07  @author John Carlos Arrieta Arrieta
08  */
09
10  package sistempres.reloj.modelo;
11
12  import java.
13
14  public class RelojModelo extends Thread {
15      // atributos
16      private
17
18      public RelojModelo() {
19      }
20
21      public void ejecutar() {
22      }
23
24      // Metodo para obtener el objeto con informacion del calendario
25      Calendar calendario = new Calendar();
26      tiempoSegundo = 0;
27      RelojModelo.getEstado() ==
28
29      RelojModelo.add( Calendar.SECOND);
30      RelojModelo.elMeridiano
31      RelojModelo.get(Calendar)
  
```

Source | Preview | JavaDoc | **Package**

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



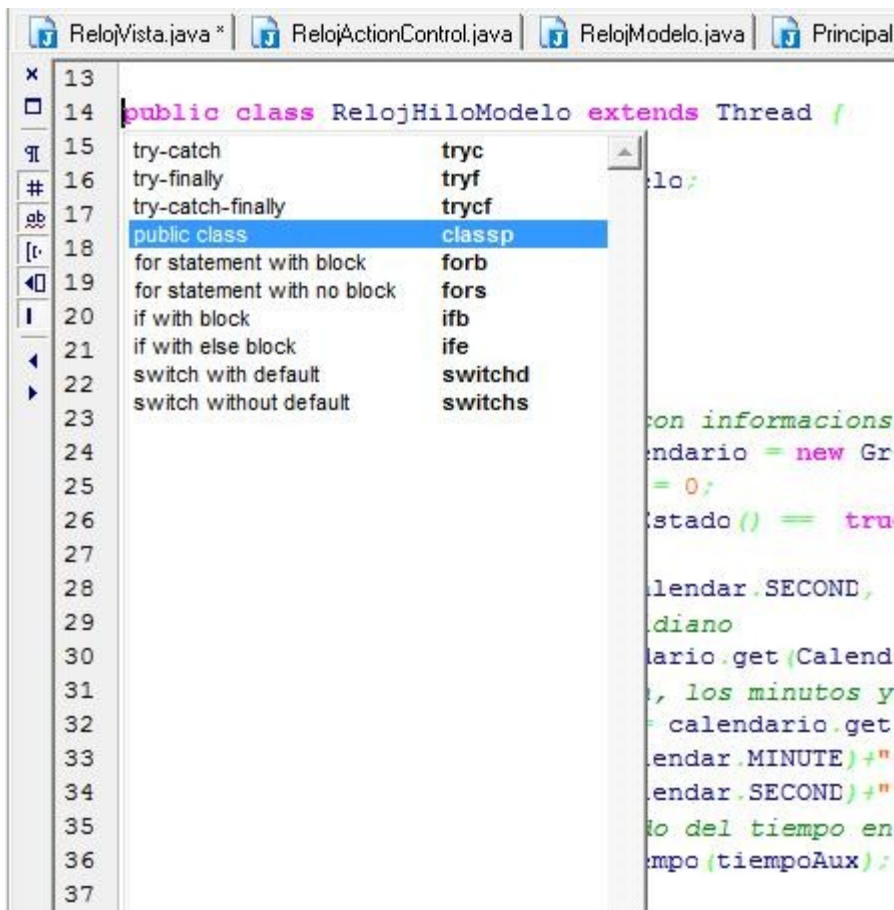
```

12 import java.util.*
13
14 public class Reloj
15 // atributos
16 private Reloj
17
18 public RelojH
19 /
20
21 public void r
22 {
23 // obt
24 Gregor
25 int in
26 while {
27 {
28 ca
29 //
30 St
31 //
32 St
33 calendario.get(Calendar.MINUTE)

```

De igual forma podemos tener autocompletado de las palabras claves y estructuras de código más utilizadas durante el desarrollo, oprimiendo la combinación de teclas SHIFT + ESPACIO, tenemos de forma automática una lista emergente de estas palabras

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



The screenshot shows the GEL IDE interface with four tabs: 'RelojVista.java *', 'RelojActionControl.java', 'RelojModelo.java', and 'Principal'. The 'RelojModelo.java' tab is active, displaying the following code:

```

13
14 public class RelojHiloModelo extends Thread {
15     try-catch          tryc
16     try-finally        tryf
17     try-catch-finally  trycf
18     public class       classp
19     for statement with block  forb
20     for statement with no block  fors
21     if with block           ifb
22     if with else block      ife
23     switch with default    switchd
24     switch without default switchs
25
26     con informaciones
27     endario = new Gr
28     = 0;
29     Estado() == tru
30
31     alendar.SECOND,
32     diano
33     lario.get (Calend
34     , los minutos y
35     = calendario.get
36     alendar.MINUTE)+ "
37     alendar.SECOND)+ "
38     lo del tiempo en
39     empo (tiempoAux);
  
```

A dropdown menu is open over the code, showing a list of Java keywords and constructs. The 'public class' option is highlighted, and 'classp' is also visible in the list.

Si en algún momento se nos olvido el nombre de algún método (función) o atributo (variable) de alguna clase, o simplemente deseamos explorar los elementos que contiene una clase, colocamos punto después de su referencia (nombre de variable de un tipo de clase) y GEL muestra una lista desplegable que contiene los elementos de dicha clase, muestra también la documentación asociada a dicho elemento, si este está documentado con el estándar Java Docs.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

```

29
30 // obtengo el meridiano
31 String M = (calendario.get(Calendar.AM_PM) == 1) ? "PM"
32 // obtengo la Hora, los
33 String tiempoAux = cale
34 calendario.get(Calendar
35 // cambio el estado del
36 relojModelo.setTiempo(t
37 try
38 {
39     Thread.sleep(1000);
40 }
41 catch (Exception e)
42 {
43 }
44
45
46
47 public void setRelojModelo(Rel
48     this.relojModelo = relo
49
50

```

java.util.GregorianCalendar

AD	int
add(int field,int amount)	void
BC	int
clone()	Object
equals(Object obj)	boolean
getActualMaximum(int field)	int
getActualMinimum(int field)	int
getGreatestMinimum(int field)	int
getGregorianCalendarChange()	Date
getLeastMaximum(int field)	int
getMaximum(int field)	int
getMinimum(int field)	int
getTimeZone()	TimeZone
hashCode()	int

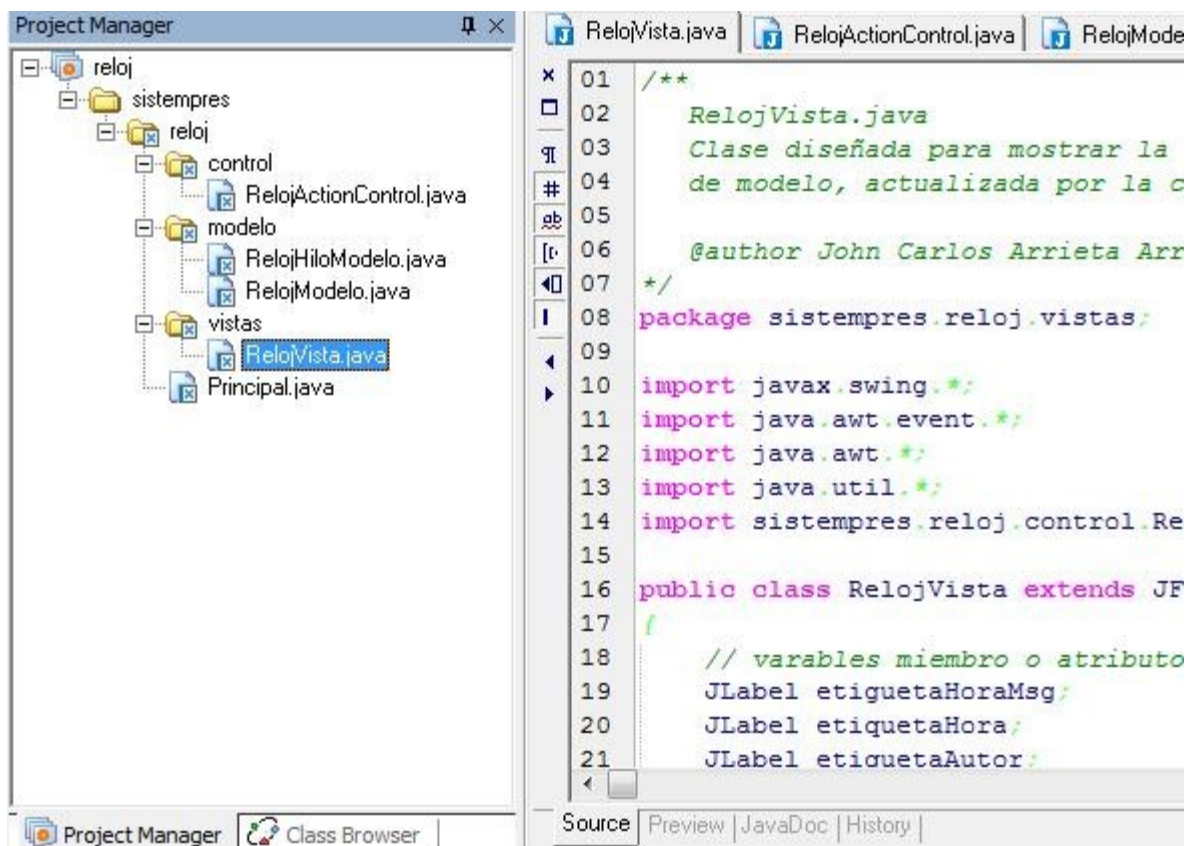
Gets the GregorianCalendar change date. This is the point when the switch from Julian dates to Gregorian dates occurred. Default is October 15, 1582 (Gregorian). Previous to this, dates will be in the Julian calendar.

Returns:
the Gregorian cutover date for this GregorianCalendar object.

La estructura del proyecto.

Una vez agregados los paquetes sus clases al proyecto, podemos ver la estructura completa del proyecto, expandiendo el árbol que se encuentra dentro del panel manejador de proyectos.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

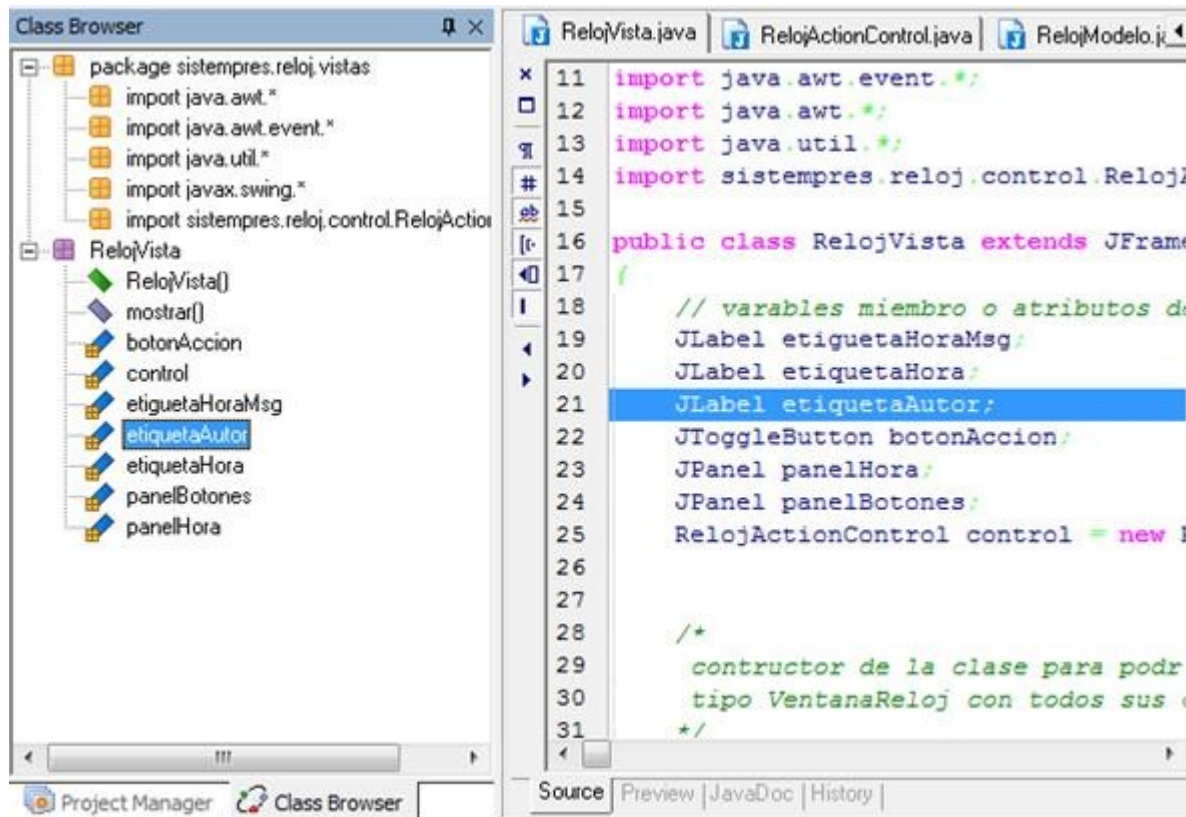


La estructura de las clases

El panel de Navegación de Clases nos permite ver en forma resumida todos los elementos que conforman el código de una clase, de esta manera ahorramos tiempo buscando el o los elementos deseados dentro del editor de código fuente. Para navegar por el contenido de una clase elemento

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

a elemento, se debe seleccionar con 2clic en el panel manejador de proyecto el ítem que corresponde a la clase que deseamos navegar, luego se selecciona con un clic la pestaña Navegador de Clases que se encuentra en la parte inferior del panel lateral desplegable, junto al panel manejador de proyectos

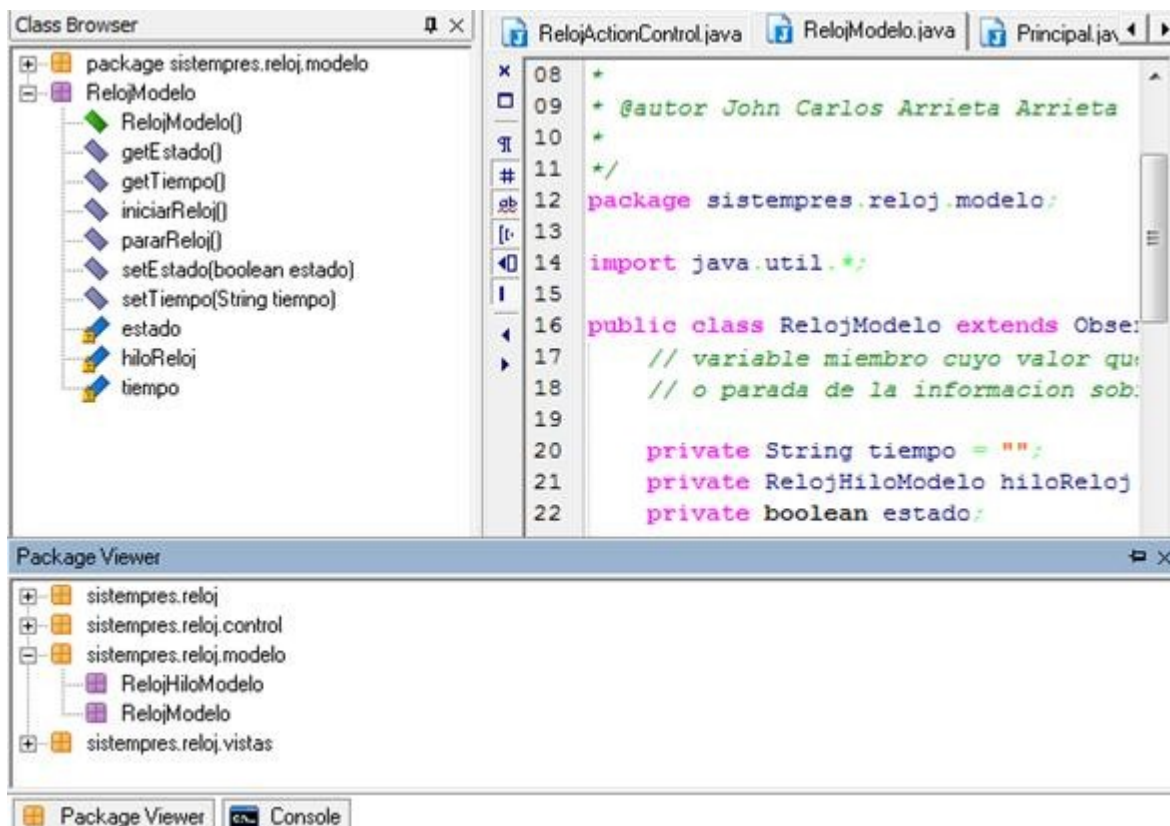



Ahora seleccionamos con 2clic cualquiera de los elementos que queremos ver dentro de la clase y GEL ubica dicho elemento dentro del código de la clase y los selecciona haciéndolo visible para nosotros.

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Navegar por la jerarquía de paquetes

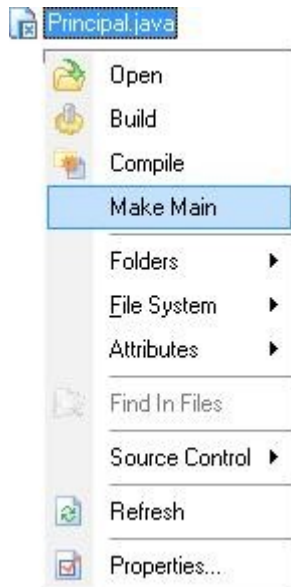
El panel de navegación de Paquetes permite ver el contenido de los paquetes que estructuran nuestro proyecto, muestra la información en forma de árbol pudiéndose apreciar la jerarquía de paquetes de nuestro proyecto



 <div> UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena </div> <div> PIENSA EN GRANDE Piensa en UNISINÚ </div>	
Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

Estableciendo el proyecto como Principal

Ahora tenemos que decirle a GEL que la clase Principal.java será la clase que inicie el programa, para ellos damos clicD sobre el archivo de la clase Principal.java, seleccionamos **Marcar como Principal**



Compilando y ejecutando el Proyecto

Una vez hecho estos pasos podemos compilar el proyecto y luego ejecutarlo, esto se hace utilizando los botones **Compilar Proyecto** y **Ejecutar Proyecto** ubicados en la barra de herramientas, también seleccionado el menú **Construir >> Depurar Proyecto** o >>> **Correr proyecto**

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



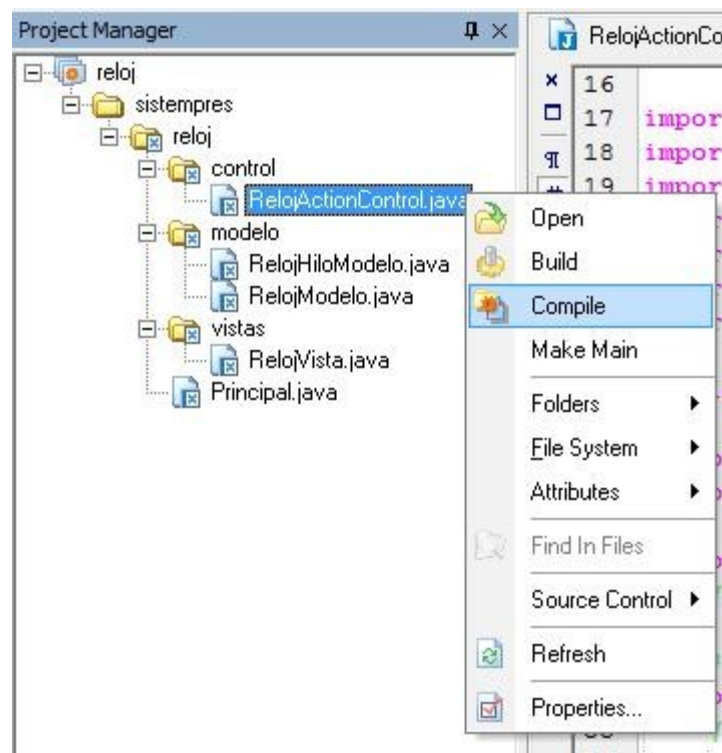
Hay tres opciones de compilación, una para compilar el proyecto completo (todas sus clases una por una), otra para compilar un paquete completo (solo toda las clases del paquete, una por una) y la tercera es la compilación por clase (se compila solo una clase y todos las clases del mismo proyecto que sean referenciadas dentro del código de la clase a compilar).

Para compilar se puede hacer de cualquiera de las siguientes formas:

>> Menu Construir >> y escoger la opción que deseamos realizar.

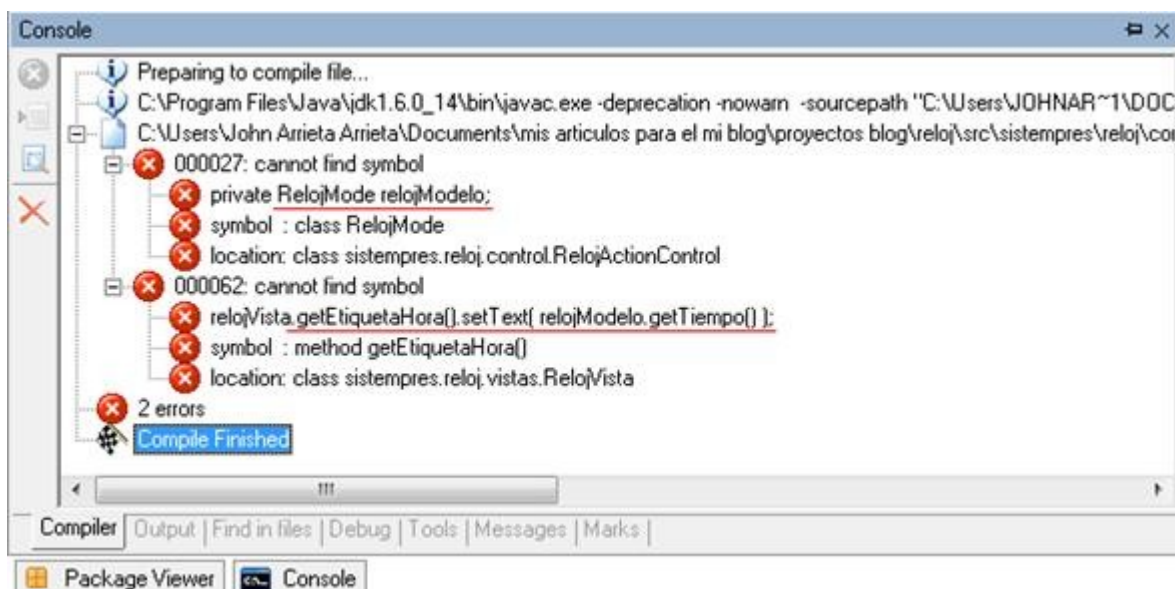
>> ClicD sobre un ítem (paquete Raíz = proyecto, un paquete o una clase) del panel manejador de proyectos y seleccionar la opción Compilar

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Sea cual sea la opción de compilación que nosotros seleccionemos, GEL llama al programa JAVAC.EXE incluido en el JSDK y le pasa los paramemos necesarios para realizar nuestra compilación, una vez terminado el proceso GEL obtiene el resultado de la compilación realizada por JAVAC.EXE y lo muestra en el panel de Consola que se encuentra en la parte inferior del editor de código y del panel manejador de proyectos

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

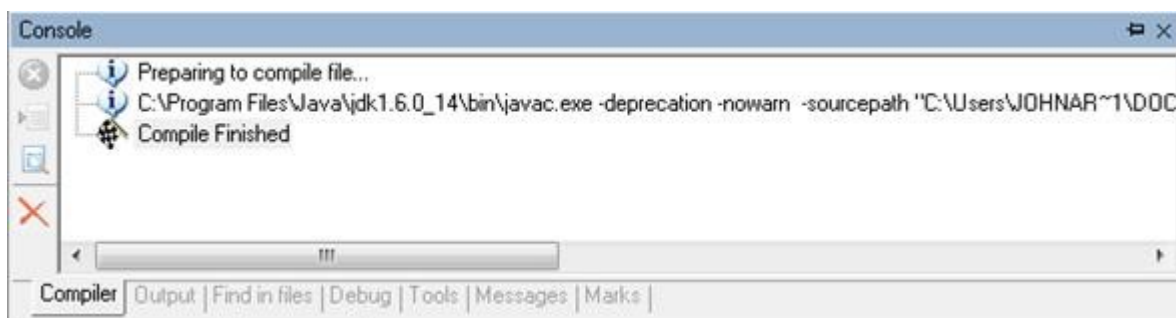


Si hay errores como en se puede apreciar en la anterior figura, GEL organiza los errores y los muestra con el símbolo rojo estándar de error, mostrando toda la información que genera el reporte de errores que genera el compilador JAVAC.EXE, pero además GEL permite ir a la línea donde se presenta dicho error, con solo dar 2 clic doble la línea subrayada que se muestra en el reporte de error del panel de Consola.

Al final del reporte GEL muestra en cifra numérica la cantidad de errores encontrados durante el proceso de compilación.

Una vez corregidos todos los errores, volvemos a solicitar a GEL que realice el proceso de compilación y el reporte satisfactorio de compilación se muestra como se ve en al siguiente figura

Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los conocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



K le pasa como parámetro el nombre de la clase Principal.java y todos los ficheros que se encuentran dentro del proyecto, esto si todo está bien y libre de errores de sintaxis en el código fuente, entonces se generaran los respectivos archivos compilados (de código binario bytecode) y se colocan en las carpetas que indica la línea de código que contiene la palabra **package**.

Enlaces relacionados:

IDEs para Java

<http://www.jedit.org>

<http://www.freealts.com/index.php>



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	

<http://notepad-plus.sourceforge.net/uk/site.htm>

<http://www.jetbrains.com/idea/nextversion/index.html>

<http://www.dcs.ed.ac.uk/home/java/about-IDE.html>

MVC

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Patrones de diseño

http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o

http://es.wikipedia.org/wiki/Observer_%28patr%C3%B3n_de_dise%C3%B1o%29

Programación concurrente

<http://nekan.net/threads-en-java-subprocesamiento-multiple>

<http://www.reloco.com.ar/prog/java/threads.html>

http://www.wikilearning.com/tutorial/tutorial_de_java-estados_de_un_thread/3938-144

<http://www.blog.qui-mera.net/?p=26>

Interfaces graficas y eventos Java para software Desktop

<http://www.programacion.com/java/tutorial/swing/>



Curso: Como construir Software escrito en Java usando la herramienta GEL	Escuela: Ingeniería de Sistemas
Guía de Aprendizaje: <ul style="list-style-type: none"> • IDE (Entono Integrado de Desarrollo) GEL 	N°: Guía Numero 27
Guía diseñada por: John Carlos Arrieta Arrieta	Tiempo: 1 HTP y 2 HTI
Competencias: <ul style="list-style-type: none"> • Comprender los cocimientos básicos del patrón de arquitectura de software MVC • Comprender los conceptos básicos del patrón de diseño de software Observer y Observable • Entender como se puede implementar una aplicación con arquitectura MVC usando el patrón de diseño Observer y Observable • Explorar y practicar con el IDE Gel, como herramienta opcional de Software libre para construir software escrito en Java. 	



Segundo articulo de Recordando el presente, Agradeciendo al Pasado, en este articulo presento como trabajar con el IDE libre GEL, con Hilos y el Patron Arquitectónico MVC para el desarrollo de software