

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)**

**Formato para prácticas de laboratorio**

CARRERA	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
Ingeniero en Computación	2009-1	12099	Programación Orientada a Objetos

PRÁCTICA No.	LABORATORIO DE	Programación Orientada a Objetos	DURACIÓN (HORAS)
9	NOMBRE DE LA PRÁCTICA	Paquetes y documentación en Java	2

**1. INTRODUCCIÓN**

En esta practica se aplicará la creación y uso de paquetes. Además se insertarán etiquetas en las clases que han modificado durante las practicas anteriores para generar la documentación.

**2. OBJETIVO (COMPETENCIA)**

Crear paquetes dentro del entorno Netbeans para que los puedan incluir en los siguientes proyectos e incluir las etiquetas necesarias para generar la documentación de las clases demostrando una actitud creativa y responsable.

**3. FUNDAMENTO**

**Paquetes.**

Un paquete es un espacio de nombres que organiza las clases e interfaces que están relacionadas. Es el mismo concepto que el sistema de archivos, en donde existen directorios o folders para organizar los archivos. Entonces, los paquetes son la forma que utiliza Java para organizar las clases e interfaces.

La plataforma Java provee de una enorme biblioteca (conjunto de paquetes) que es adecuado para nuestras aplicaciones. A esta biblioteca se le conoce como *API (Application Programming Interface)*.

Los creadores de Java hacen algunas recomendaciones para nombrar los paquetes que es:

- 1.- Comenzar con letra minúscula.
- 2.- La compañías utilizan su dominio de Internet pero de forma inversa para nombrar sus paquetes, por ejemplo `com.ejemplo.orion.espacio_de_nombre`
- 3.- Los paquetes del lenguaje Java comienzan con **java.** o **javax.**

Para utilizar una clase de un paquete se puede hacer con:

```
graphics.Rectangle miRectangulo = new graphics.Rectangle();
```

o también importando el archivo:

```
import graphics.Rectangle;
```

### Comentarios de documentación.

Los comentarios de documentación una visión de más alto nivel para gente que no tiene el código a mano y que sólo quiere usarlo. El JDK proporciona una herramienta para generar páginas HTML de documentación a partir de los comentarios incluidos en el código fuente. El nombre de la herramienta es **javadoc**. Para que javadoc pueda generar los textos HTML es necesario que se sigan unas normas de documentación en el fuente, que son las siguientes:

- Los comentarios de documentación deben empezar con `/**` y terminar con `*/`.
- Se pueden incorporar comentarios de documentación a nivel de clase, a nivel de variable (dato miembro) y a nivel de método.
- Se genera la documentación para miembros `public` y `protected`.
- Se usan etiquetas (*tags* en inglés) para documentar ciertos aspectos concretos como listas de parámetros o valores devueltos. Algunas etiquetas comúnmente usadas se describen en la siguiente tabla.

Donde aplica la etiqueta	Etiqueta	Descripción
Todos	@see	Permite crear una referencia a la documentación de otra clase o método.
Clases	@version	Número de versión de la clase.
Clases	@author	Nombre del autor.
Clases	@since	Fecha desde la que existe la clase.
Métodos	@param	Parámetros que recibe el método.
Métodos	@return	Significado del dato que devuelve el método.
Métodos	@throws	Comentario sobre las excepciones que lanza.
Métodos	@deprecated	Indica que el método es obsoleto.

Toda la documentación del API de Java está creada usando esta técnica y la herramienta

javadoc. Se puede ver información adicional sobre javadoc en:

<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

**Importante:** No se pueden usar comentarios de documentación para bloques de código dentro de métodos porque javadoc los asigna a la primera declaración que encuentre detrás de ellos

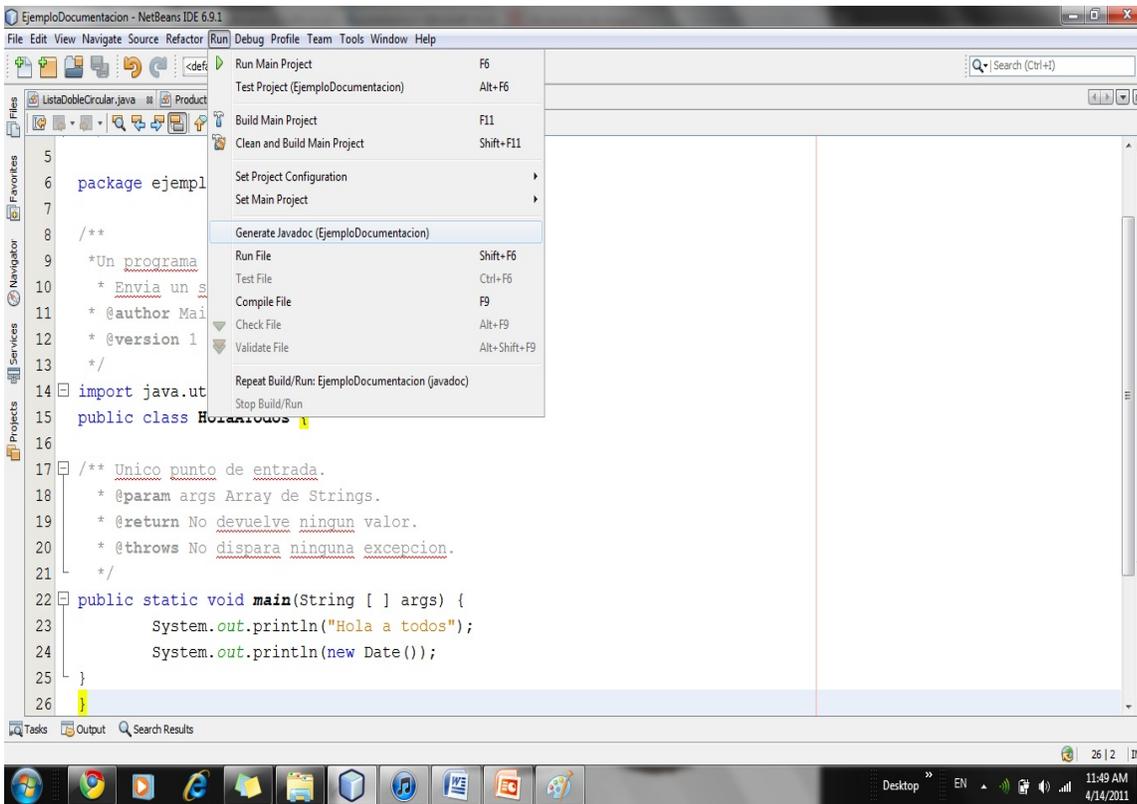
Por ejemplo

```
import java.util.*;

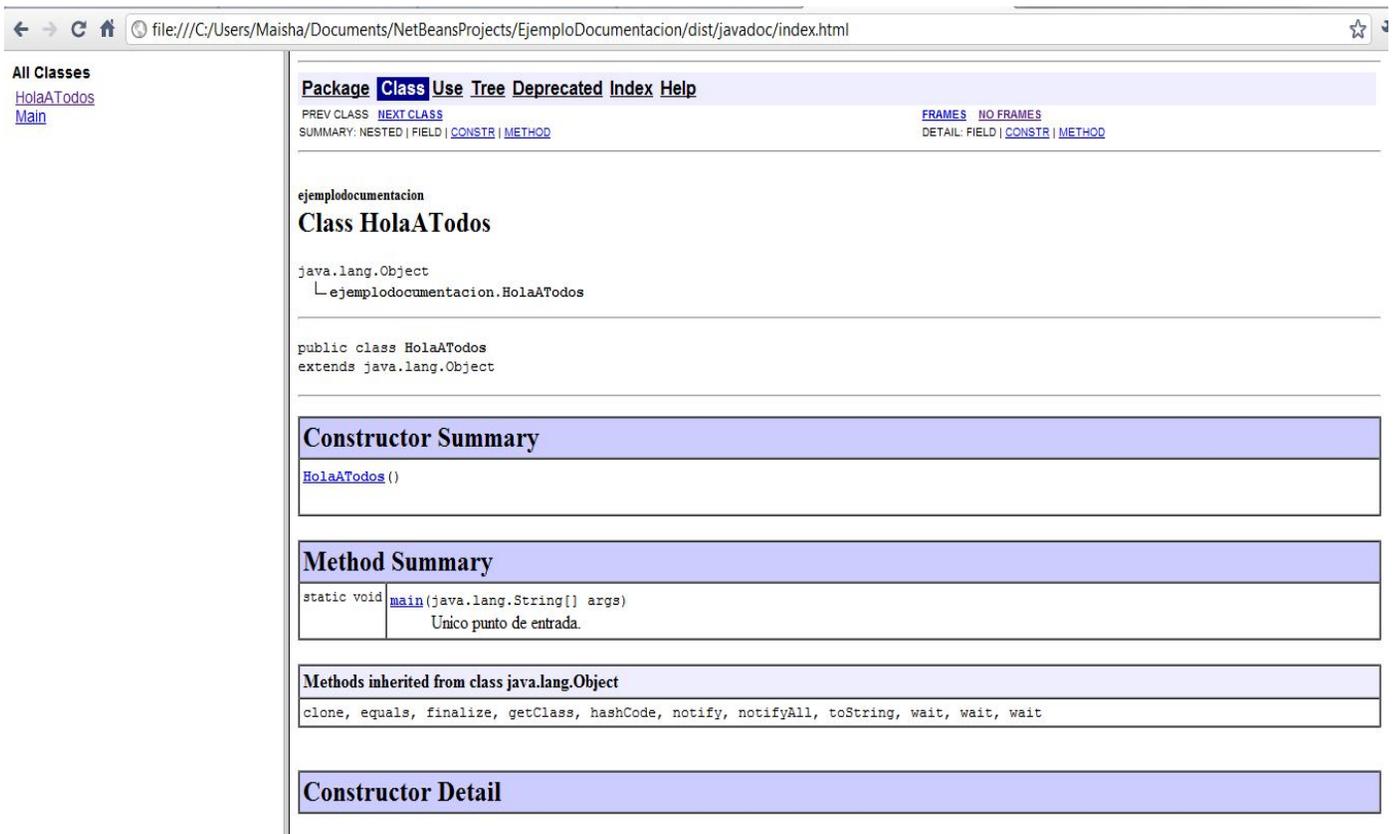
/** Un programa Java simple.
 * Envía un saludo y dice que día es hoy.
 * @author xxxx
 * @version 1
 */
public class HolaATodos {

    /** Unico punto de entrada.
     * @param args Array de Strings.
     * @return No devuelve ningun valor.
     * @throws No dispara ninguna excepcion.
     */
    public static void main(String [ ] args) {
        System.out.println("Hola a todos");
        System.out.println(new Date());
    }
}
```

Para que se genere la documentación en Java debemos seleccionar del menú principal de Netbeans, la opción de Run y después la opción Generate Javadoc, como se muestra en la imagen.



El resultado se muestra en la siguiente figura. Como se puede observar se generan páginas HTML con la misma apariencia y con la misma organización que la del API de Java.



#### 4. PROCEDIMIENTO (DESCRIPCIÓN)

- 1.- En la práctica 8 creo una clase que lee desde la línea de comandos. ¿Qué cambios deberá hacer a esta clase para que forme parte en el paquete `misClases`?
- 2.- Utilizando las etiquetas de javadoc explicadas anteriormente, documente las clases que fueron modificadas en las practicas anteriores, como por ejemplo las clases `Picture` y `Turtle`. Utilice las etiquetas adecuadas para poner comentarios en los métodos que usted ha creado y agregado a dichas clases.
- 3.- Genere los archivos con la documentación de las clases y verifique su contenido utilizando un navegador.
- 4.- ¿Qué ventajas tiene la herramienta javadoc?
- 5.- ¿Qué ventajas tiene el organizar las clases en paquetes?

#### A) EQUIPO NECESARIO

#### MATERIAL

Computadoras con capacidad para ejecutar el entorno de desarrollo Netbeans. Paquete `misClases`.

#### 7. REFERENCIAS

Netbeans

<http://netbeans.org/downloads/>

Java 6

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Formuló Cecilia Curlango Rosas Maria Luisa Gonzalez Ramirez	Revisó Gloria E. Chavez Valenzuela	Aprobó	Autorizó M.C. Maximiliano de las Fuentes Lara
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Responsable de Gestión de Calidad	Nombre y Firma del Director de la Facultad