**Clase 01 – Programador Java**

**Introducción – Clase 01 Java**

Las famosas IDE:

* Netbeans (SUN)
* Eclipse (IBM): El más usado en las empresas
* JBuilder, Jcreator
* Intellij Idea (El mejor IDE, pero es pagado XD, se ejecuta en android)
* **BlueJ**

Ver lo que se necesita como analista programador:

* Bumeran – Java.

Libro de Patrones

* La banda de los 4: John Ralph, Erick, Richard

Usar Git: Registrar, Instalar, Clonar, hacer un cambio a mi repositorio.

* Curso de Git: Udemy.com
* Sw de ayuda: Zomit
* Crear tu GIT en github: repositorio PROGRAMADOR\_JAVA / gustavopajueloreb@gmail.com
* Descargar GIT for Windows: Git-2.7.0-64-bit
* Github del profe: https://github.com/gcoronelc/SISTUNI\_PROG\_JAVA\_002
* Instalar Git for Windows
  + Seleccionar al Path
  + Seleccionar comando de Windows
* Entrar a consola y entrar a la carpeta EGCC
* Ejecutar c:\EGCC>git clone <https://github.com/gcoronelc/SISTUNI_PROG_JAVA_002.git>
* Ejecutar c:\EGCC>git clone <https://github.com/gustavopajueloreb/PROGRAMADOR_JAVA.git>
* Entrar en consola a la carpeta del ultimo clonado.
* Editar readme.md
* Usar comando “git status” ver que se modifico.
* Usar comando “gist add .” esta preparando a todos los archivos para hacer commit
* Usar comando git commit -m "pucha que uso git weon"
* No estas logueado :c
* Logueate:
  + git config --global user.email [gustavopajueloreb@gmail.com](mailto:gustavopajueloreb@gmail.com)
  + git config --global user.name "gustavopajueloreb"
* Ejecuta de nuevo el commit: 1 file changed
* Ejecuta: “git push origin master”, ingresa tu usuario y contraseña de github

Movimientos fundamentales del GIT

* Clone <\_
* Push \_>
* Pull <\_

Cargar cambios hechos en repositorios de otros:

* Comando para retroceder carpeta “cd ..”
* Entras a SISTUNI\_PROG\_JAVA\_002
* Ejecuta: git pull origin master

Ejemplo a seguir

* Paul Graham, tiene blog

Convenciones de Java y buenas prácticas

* Java Code Conventions

Desarrollo de Proyectos de SW

* SCRUM, RUP, PRINT.

SOA

* Hay especialistas en temas de integración SOA, integración de sistemas como servicio

ESB

* También es de integración

Principios del programador

* Hay que establecer los principios del programador
* Libro Piam code

Conoce gente nueva

* YO S.A., libros de Ines Temple

Repositorios llenos de información en Java, clases y videos

* https://github.com/gcoronelc/PECI-Java-MAR-2015

Frase motivacional

* “Yo entendí esto, entendí que usted entiende que yo entiendo esto y usted entiende que yo entendí esto. Nos entendemos.”
* Explicación la Eqitacion entre el Project manager y los stakeholders durante el proceso de los requerimientos.

En los bancos y seguros el desarrollo de SW pasa por las siguientes áreas.

* Área de desarrollo, área de QA, área de certificación, área de preproducción y área de producción.

¿No te gusta hacer pruebas de tu SW?

* <http://elcomercio.pe/economia/personal/tu-cuenta-interbank-cero-banco-promete-pronta-solucion-noticia-1863970>

Un programador compra MAC no laptop

* Está en un sistema UNIX (Linux)

**NETBEANS – Clase 01 Java**

* La raíz del proyecto se guarda en la carpeta del proyecto “src”
* El nombre del package debe respetar la convención de nombrado de paquetes.
* Si quieres cambiar el nombre de una clase junto con el nombre del archivo java es obligatorio en ambos, ejecuta “Refactor -> Rename” o Ctrl + R.
* Operaciones del Objeto -> Métodos de la clase, son servicios proporcionados por un objeto
* Durante la documentación del archivo .java los comentarios y la estructura d estar documentado igualmente a “Documentacion tipo JAVA DOC”
* Objeto es la instancia de una clase:

Trabajador obj = new Trabajador();

Operador que crea el objeto

Trabajador obj1 = obj;

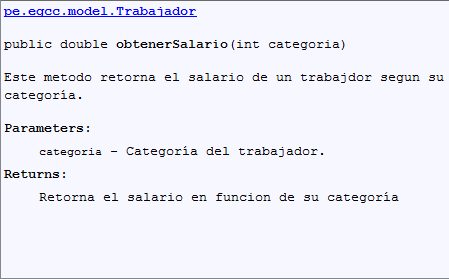
Objeto

Obj

Apuntador

Obje1

* Cuando cumples con la documentación JAVA DOC al presionar ctrl + barra sobre el Objeto o el método usado, entonces te menciona la información del sudodicho, información limpia y congruente.



* Para hacer comentarios de la clase escribe /\*\* y enter.

**Proyecto Práctico – Clase 01 JAVA**

Identificar los servicios.

1. Calcular el impuesto a partir del pedido
2. Calcular el importe total

Jamás hagas cálculos u operaciones en las vistas de usuario, capa de usuario. Todo relacionado a los procesos del negocio se hace en la capa de servicios.

Esto logra escalar el proyecto, y subir a otros niveles.

Toda aplicación empieza a ejecutarse con una clase principal.

Cuando creas el proyecto PedidoApp hay que configurar la clase principal:

* pe.egcc.pedidoapp.ClasePrincipal

paquete base clase principal

Es bueno enfocarse en capas de tu app: View (FrontEnd), Services, prueba, y main.