# ÍNDICE

[ÍNDICE 1](#_Toc95321304)

[1. Estructura de un programa en java con eclipse 2](#_Toc95321305)

[2. Crear una carpeta en eclipse 2](#_Toc95321306)

[3. Archivos properties en eclipse 3](#_Toc95321307)

[4. Como invocar a un archivo que esta dentro de la carpeta resources en eclipse 3](#_Toc95321308)

[5. Instalacion de selenium 3](#_Toc95321309)

[6. Script de selenium 4](#_Toc95321310)

[7. Proyecto JazztelAuto 4](#_Toc95321311)

[8. Instalacion Eclipse 10](#_Toc95321312)

[9. Contenido del POMV raíz 11](#_Toc95321313)

[10. Conceptos de Java - Jar 11](#_Toc95321314)

[11. Ejecutar un archivo .jar 12](#_Toc95321315)

[12. Descargar el archivo Chromedriver y colocarlo en el workspace (En la carpeta resources) 12](#_Toc95321316)

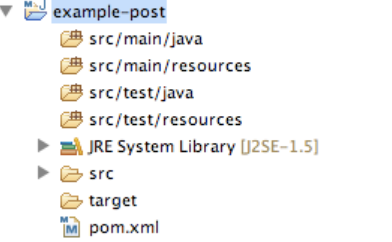
[13. Conceptos de Java – Class 12](#_Toc95321317)

[14. Conceptos de Java – Variables 13](#_Toc95321318)

[15. Conceptos de Java – Control de errores 13](#_Toc95321319)

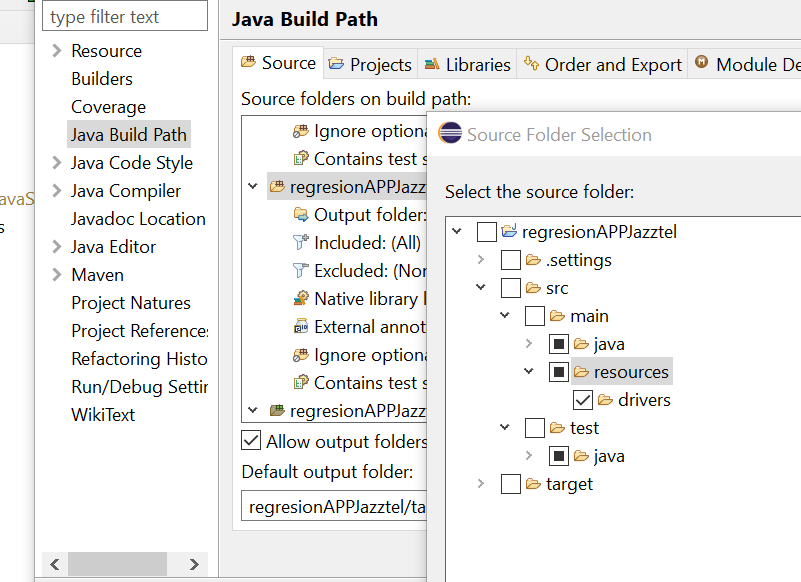
## Estructura de un programa en java con eclipse

* **src/main/java** : donde guardaremos nuestras clases java **fuente (\*.java)**. Debajo de esta carpeta situaremos nuestras clases en distintos paquetes.
* **src/main/resources** : aquí almacenaremos los recursos (ficheros xml, ficheros de propiedades, imagenes, …) que pueda necesitar las clases java de nuestro proyecto.
* **src/test/java** : en dicha carpeta se guardan las clases de test que se encargarán de probar el correcto funcionamiento de nuestra aplicación. Aquí por ejemplo podemos guardar nuestros test unitarios de **JUnit**.

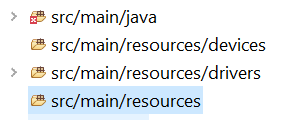


## Crear una carpeta en eclipse

Para crear una carpeta en eclipse nos vamos a properties del proyecto y en java build path pestaña sources le damos a Addfolder.



Esto es necesario para no meter todo a mogollon en la carpeta resources. Luego en eclipse no se vera con estructura de árbol sino asi:



## Archivos properties en eclipse

## Como invocar a un archivo que esta dentro de la carpeta resources en eclipse

## Instalacion de selenium

Los archivos necesarios se encuentran en esta pagina Web:

http://www.seleniumhq.org/download/

## Script de selenium

Se debe agregar la dependencia de selenium-java en el archivo **pom**.xml para que funcione el script de selenium.

## Proyecto JazztelAuto

**NOTA**: IMPORTANTE

- eclipse

- por defecto se abre el proyecto jazztel

- hay cuatro carpetas: cliente ws

- azztel auto. es el proyecto padre

- jaztel crm. este es nuestro proyecto a usar

- jazztel local. Es un proyecto no subido

- Maven lo utilizamos con Cilantrum para establecer una estructura común de directorios

Una de las herramientas más útiles a la hora de utilizar librerías de terceros es Maven.

Maven se utiliza en la gestión y construcción de software. Posee la capacidad de realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado. Es decir, hace posible la creación de software con dependencias incluidas dentro de la estructura del JAR.

Es necesario definir todas las dependencias del proyecto (librerías externas utilizadas) en un fichero propio de todo proyecto Maven,

el POM (Project Object Model). Este es un archivo en formato XML que contiene todo lo necesario para que a la hora de generar el fichero

ejecutable de nuestra aplicación este contenga todo lo que necesita para su ejecución en su interior.

Sin embargo, la característica más importante de Maven es su capacidad de trabajar en red. Cuando definimos las dependencias de Maven, este sistema

se encargará de ubicar las librerías que deseamos utilizar en Maven Central, el cual es un repositorio que contiene cientos de librerías constantemente

actualizadas por sus creadores. Maven permite incluso buscar versiones más recientes o más antiguas de un código dado y agregarlas a nuestro proyecto.

Todo se hará de forma automática sin que el usuario tenga que hacer nada más que definir las dependencias.

El proyecto esta en la direcion: https://slsvn.indra.es/JAZZTELSCRM/automatizacion\_pruebas/automatizacion/trunk/JazztelAuto

- (PEDIR PERMISOS DE USUARIO PARA SVN, PARA LA DIRECION ANTERIOR, SINO NO PODRE hacer commit ni descargar de subversion)

- dentro de jazztel crm - toda la parametrizacion esta en recursos

- dentrod e resources en sgl estan los scripts -generar usuarios,

para la direccion vamos al scripts de crear cliente fibra jazztel. sql

antes de esto e crm direccion creaos un registro con nuestra nueva direcion, ya que sino el script de creacion de cliente no hace nada

en la parte de direcion del sql, tene en cuenta que no hay que meter direcciones con ñ, sino el sql peta. Mejor meterla sin ñ y luego gescalizarl

- trasladamos los datos de crm direccion al sql. (copy paste)

- al crear la direccion en crm se crea registro en crm direccion y de ahi copiamos los datos en nuestro scrip

- utilizar el lanzarautojaztel, este lanza en it, los otros dos se generan en pre i jekkinds

. se selecciona la suite recursos dentro del java

se hace un control + mayusc + o

y se importa el recurso

nos vamos a la suite recursos - dentro de suits recursos - descomentamos el de tv

abrimos dentro de recursos el fibra jazztel tv

- mavel controla las dependencias apache del proyecto (repositorio Oracle, apache, mavel). El svn solo controla los archivos del proyecto

es la que está ahora con auto

Santos Peña, Estela María <emsantosp@minsait.com>

- apache asis controla la conexion con los web services

- lanzar auto jazztel. se laza en it

- dentro estan las suite de regresion

- las funciones tiran de losexcel como datos tcdata.xls

- para cambar los datos de las pruebas se cambian los excel.

- las paginas web estan metidas en el archivo webs\_properties

- los java se compilan con cilantrum

- desde el auto java, se selecciona la suite, dentro de la suite los casos de prueba que se quieran

- en el excel caa pestaña es unnombre de suite

- en la st\_recursos se

- la columna numero de recursos es restarle uno

- en el excel se meten los datos de la prueba

- el archivo valores ajuste .txt son valores de referencia basicos para algu nos casos de prueba

- para lanzar caso de prueba, import suite recursos (se da play desde aqui, desde el main), descomentar caso de prueba, excel datos

tener permisos de administrador en la maquina

- las suites cuelgan de src

primero esta el com.main- hay tres main

mas abajo estan el resto - pantallas - modela cada pantalla de crm

- los reutilizables, es el motor del proyecto

- las suites son las ue se lanzan (generar, datos, regresion)

- el log--primero pusamos f5 para actualizar el proyecto y vamos a la carpeta logs

- el reporte de resultados esta en jazztel pedro - navegamos por las carpetas hasta proyectos / proyecto jazztel/testreport

- la aplicacion excel debe estar cerrada mientras se ejecute el programa

- el test report solo se abre con firefox

- el excel esta donde esta el xecel de la prueba

- dentro deresources esta el excel . la direcion se mete en los dos sitios, excel y pl de cliente

- el pl lo coge la funcio crear cliente de cada suite

- resourcesjazztelcrm-tc data aqui esta el excel

- los equipos estan en la excel

- si se quiere generar varios recursos se buclar el lanzar auto

- si se ve que va lento matar el explorer

taskill /F /IM IEDriverServer.exe

esto es por un fallo de cilamtrum que no mata las tareas del explorer

--------------

Pruebas de regresion

hay colgando de src una suite de regrsion, dentro de la cual estan caa una de las funciones

dentro de la suite auto estan las reutlilizables, que son funciones propias generales de crm

el apache maven esta el cnf configurado para pablo y pedro , entonces hay que ir al directorio, tocar el dnf

cambiar el settings de pablo por el de pedro Dirección C:\Program Files\apache-maven-3.5.2\conf

la libreria de los ws de crm la manda por correo, ya que a veces no se referencia bien. A la hora de montar el proyecto

construirlo en maven - email Dependencia rara del WS de CRM

- el web properties contiene las direcciones de los sistemas asociados a crm.

- meter en el explorer como excepcion el repositorio de indra

- ficha tiene su propio suite de reutilizables y su propia suite de regresion (en el excel se referencia a ellos)

- distribuidores tiene tambien su prooia suite degresion

ojo, auque los casos de distribuidores fallen se crean bien en crm

- cada pantalla de cada sistema tiene su propia suite de pantalla

- luego en reutilizabes tambien hay una parte propia de cada sistema, como wd

- las llamadas a crones tambien estan automatizadas dentro del java

- en el java dentro de reutilizables, herramientas crm estan las llamadas a los rfs, cada uno tiene una funcion

- los pantallazos a adjuntar en el quality center salen en el test report

- en el caso de prepago, de los verificadores, hay subrutina de verificacion dependiendo de si hay prepago o no

- para dar ordenes concretas a selenium se utilizan funciones de cilantrum. Se pueden programar el cilantrum para nuevas funciones

que indiquen a selenium lo que tiene que hacer

- en la suite de auto crm de acciones especiales estan metidas funciones de acceso a luna, las cuales invocan directamente a codigo selenium

- todo se ejecuta en internet explorer

- dsde que sale el pedido de crm todo el resto (luna, etc) se hace en una misma funcion

- pedir permiso svn repositorio a sergio

- dentro del eclipse se puede ir creando una suite a nivel local para lanzar pruebas concretas

- en el excel si se trata de una onte integrada (router premium) se deja el campo router a blanco (todos los campos del excel

son entradas para las pruebas a ejecutar)

- a la hora de cerrar las ot se comprueba si ficha funciona, y si no funciona se hace tuneandolo por medio de los xml

- dentro de resources, valores\_ajuste.txt - este archivo contiene valores unicos para contratar, como numeros de movil o fijo pvoo. En cada

ejecucion el procesoaumenta esos valores

- en la suite de recursos, en generar el pat, hay una parte 5 de generar movil. Se puede comentar el resto e ir directamente a la parte movil

- ctrl+7 comenta un bloque entero de codigo

---------------------------

- JUNIT. JUnit nos permite, a través de multitud de etiquetas,

organizar esa batería de pruebas, para ejecutarlas y obtener los reportes de manera automática.

https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/385

IMPOTANTE

---------

Se monta sí, tenemos maven - repositorios git - jenkins para que se ejecuten las pruebas unitarias

, los procesos

es que el jenkins lo tenemos enlazado con git

cuando tu haces un commit en la rama develop por ejemplo

automáticamente se ejecuta en jenkins un job de build de ese micro

pero esos temas de conexiones entre git y jenkins no los manejamos nosotros

## Instalacion Eclipse

https://ebooksonline.es/integracion-de-maven-jenkins-con-selenium-un-tutorial-completo/

https://qastack.mx/programming/3724415/maven-artifact-and-groupid-naming

- Maven se utiliza para definir la estructura del proyecto, las dependencias, la construcción y la gestión de pruebas.

- Con pom.xml (Maven) puede configurar las dependencias necesarias para la compilación actual y las pruebas de código.

- Maven descarga automáticamente los archivos necesarios del repositorio mientras se construye el proyecto.

para lanzar las pruebas es necesario ejecutar el servidor de Selenium, que es el que se va a encargar de ejecutar las órdenes que se le indiquen a través de las pruebas grabadas. Para ello se descarga de la página oficial el Selenium RC,

simplemente hay que localizar el fichero “selenium-server.jar” y arrancarlo con “java -jar selenium-server.jar”.

Crear proyecto junit selenium

https://5balloons.info/write-selenium-test-cases-with-junit-maven-and-eclipse/amp/

## Contenido del POMV raíz

https://es.myservername.com/use-maven-build-automation-tool

* **groupId** identificará su proyecto de forma única en todos los proyectos, por lo que debemos aplicar un esquema de nombres.
* **artefactId** es el nombre del jar sin versión. Si lo creó, puede elegir el nombre que desee con letras *minúsculas* y sin símbolos extraños.

## Conceptos de Java - Jar

Un archivo .jar suele albergar una **biblioteca con varios archivos**. La extensión .jar es una abreviatura de**Java Archive** y suele contener, como su nombre sugiere, varios archivos Java y metadatos que **se envían de manera sintetizada y comprimida.**Además de archivos .jar, esta extensión puede contener también imágenes, archivos de audio u otros formatos y funciona de manera similar a un archivo .zip.

Uno de los componentes más importantes de estas bibliotecas de Java es el llamado **manifiesto**, un archivo que contiene detalles acerca de la versión y el autor del código

Para **acceder a los archivos albergados** en una biblioteca Java, es necesario descomprimirla y extraerlos, lo cual puede hacerse igual que si se tratara de un archivo .zip. Los siguientes programas son capaces de descomprimir el formato de archivo JAR:

* **WinRAR**es una de las herramientas gratuitas más populares para extraer archivos.
* **WinZip**está disponible en versión estándar por un económico precio y con gran variedad de opciones.
* **7-Zip**dispone de su propio y eficiente formato de archivos para bibliotecas: .7z.

## Ejecutar un archivo .jar

Si la biblioteca Java es, en sí misma, un archivo ejecutable, necesitarás un **programa especial** para abrirla. El más común es **Java Runtime Environment**(JRE).

El archivo .jar puede ejecutarse mediante un doble clic.

El JRE es un entorno en tiempo de ejecución que permite ejecutar archivos mediante una [máquina virtual](https://www.ionos.es/digitalguide/servidores/know-how/maquina-virtual/) –la Java Virtual Machine (JVM) – sea cual sea el sistema operativo.

## Descargar el archivo Chromedriver y colocarlo en el workspace (En la carpeta resources)

Una vez que hemos descargado el exe chromedriver y lo hemos metido en una carpeta drivers dentro de resources



Hacemos:

**this**.getClass().getResource("/chromedriver.exe");

## Conceptos de Java – Class

¿Qué es una clase en java?

NOTA: **cada** clase generada nos crea en el proyecto un archivo java

La estructura de una clase es:

class [nombre de la clase] {

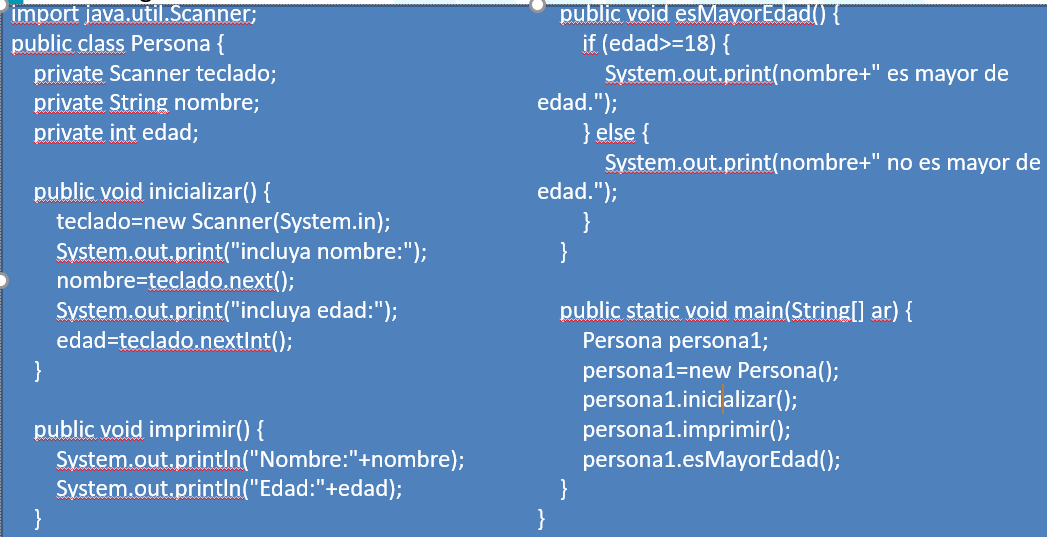
[atributos o variables de la clase]

[métodos o funciones de la clase]

[main]

}

Una clase es una plantilla (molde), que define atributos (variables) y métodos (funciones). Se definen siempre como *publicas*, ya que luego se crean instancias de ella con new.



Debemos crear una clase antes de poder crear objetos (instancias) de esa clase. Al crear un objeto de una clase, se dice que se crea una instancia de la clase o un objeto propiamente dicho. ***Instanciar*** objetos es el proceso de generar un ejemplar de una clase.

Los objetos se crean a partir de una clase. Con el nombre del objeto (o la variable que va a contener ese objeto que deseamos crear), seguido de un signo igual y luego la palabra **new** seguida del nombre de la clase y unos paréntesis.

System.***out***.println( "Hello World!" );

login milogin;

// Ahora instanciamos la fucking clase

milogin = **new** login();

## Conceptos de Java – Variables

Las variables de la **clase** se suelen crear como ***privadas***, aunque puede haber alguna publica como la de webdriver porque instancia a un método de una librería u objeto externo.

**public** **static** Webdriver.driver();

## Conceptos de Java – this

La keyword ‘**this**‘ es una variable que hace referencia al objeto actual

**this**.getClass().getResource("/chromedriver.exe");

## Conceptos de Java – archivo de propiedades

Si queremos almacenar ciertos valores predefinidos de variables generales, como las url de webs a las que vamos a invocar, defibimos archivos txt de properties.

<https://elfreneticoinformatico.com/como-establecer-archivo-de-propiedades-en-java/>

* El archivo puede tener cualquier nombre, pero la extensión siempre debe ser .properties y, como es de esperar, puedes guardarlo en la ubicación que más te convenga.
* El estándar de escritura de cada propiedad debe ser así: nombre\_propidedad=valor\_propiedad

La única norma es que no debe contener **espacios**

Los archivos se almacenan en: C:\Users\predondo\eclipse-workspace\regresionAPPJazztel\src\main\resources\**devices**

Luego en nuestra clase lo importamos:

properties.load(**new** FileReader("web.properties"));

## Conceptos de Java – Control de errores

**try** {

properties.load(**new** FileReader("web.properties"));

String urlaooent2 = properties.getProperty("url\_chrome\_app\_ent2");

*driver*.get("url\_chrome\_app\_ent2");

System.***out***.println("url\_chrome\_app\_ent2"+ urlaooent2);

} **catch** (Exception ex) {

//Si hay algún error, por ejemplo que no exista el archivo de propiedades

// se dejan los valores predeterminados del language (ES) y lastAccess (null)

System.***out***.println("Se ha producido un error al leer el archivo web.properties" );

}

Por ejemplo, para abrir Chrome:

* Que es new
* Porque las clases son publicas
* Como sabemos como importar el Chrome , su librería y sus funciones
* Como sabemos que función y librería contiene lo que queremos?

## Conceptos de Java – junit

**import** org.junit.Test;