



# INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

## MÓDULO 4: HERENCIA

### Guía de trabajo del reto de la extensión de las líneas telefónicas

---

A continuación se le presentará un reto en donde se espera que usted fortalezca sus conocimientos y habilidades para:

- Implementar y usar el mecanismo de herencia.
- Crear clases a partir de clases ya existentes.

La siguiente guía le ayudará a desarrollar el reto siguiendo un paso a paso. Podrá apropiarse del problema e implementar el código fuente necesario en el lenguaje de programación orientado a objetos Java para que el programa funcione correctamente.

#### ETAPA 1: PREPARACIÓN DEL ENTORNO DE DESARROLLO

Para preparar el entorno de desarrollo de la presente actividad, siga las instrucciones dadas a continuación:

1. Para conocer el funcionamiento esperado de la aplicación, descargue y/o ejecute el archivo demo del ejercicio, el cual puede encontrar en la siguiente lección, llamada “Demo de la aplicación final de extensión de líneas telefónicas”.
2. Descargue el archivo comprimido m4\_ejercicio\_lineasTelefonicas que contiene el proyecto.
3. Descomprima el archivo e impórtelo en el entorno Eclipse.  
Si tiene dificultades para llevar a cabo este paso, diríjase a la sección “Recursos”, ubicada en el menú principal del curso (en la parte izquierda de la ventana). Allí encontrará los “Videos de ayuda”. Siga el enlace indicado y vea el vídeo “Descargar, descomprimir e



## INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

importar proyectos a Eclipse”, que corresponda al ambiente en el que usted está trabajando (Windows o Mac). En este vídeo se explica detalladamente cómo descargar, descomprimir e importar proyectos a Eclipse.

Finalmente, limpie el ejercicio. Para esto, en Eclipse vaya a: Project > Clean > Ok. Si tiene dificultades para llevar a cabo este paso, diríjase al enlace indicado en la página “Videos de ayuda” de la sección “Recursos” y vea el vídeo “Uso de clean en Eclipse”. Este vídeo le muestra un ejemplo de cómo limpiar un proyecto en Eclipse.

**No se preocupe si algunas clases le reportan errores (como marcas rojas) al importar el ejercicio por primera vez. Estos desaparecerán cuando termine correctamente los cambios requeridos en el mundo.**

### ETAPA 2: APROPIACIÓN DEL PROBLEMA

1. Lea la descripción del problema disponible en:  
**[m4\\_ejercicio\\_lineasTelefonicas/docs/specs/Descripcion.pdf](#)**.
2. Lea el documento de requerimientos funcionales disponible en:  
**[m4\\_ejercicio\\_lineasTelefonicas/docs/specs/RequerimientosFuncionales.pdf](#)**.
3. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en:  
**[m4\\_ejercicio\\_lineasTelefonicas/docs/specs/ModeloConceptual.jpg](#)**.  
Identifique las clases, relaciones entre clases, atributos y métodos.
4. Revise el proyecto Java. Identifique los elementos del mundo (clases, atributos y métodos).
5. Se recomienda que consulte los siguientes videos, los cuales se encuentran en el enlace indicado en la página “Videos de ayuda” de la sección “Recursos”:
  - “Package Explorer en Eclipse”. Este video le explicará cómo acceder al explorador de Eclipse.
  - “Outline en Eclipse”. Este video le explicará cómo listar y acceder directamente a los métodos y atributos de una clase en forma rápida y eficiente.



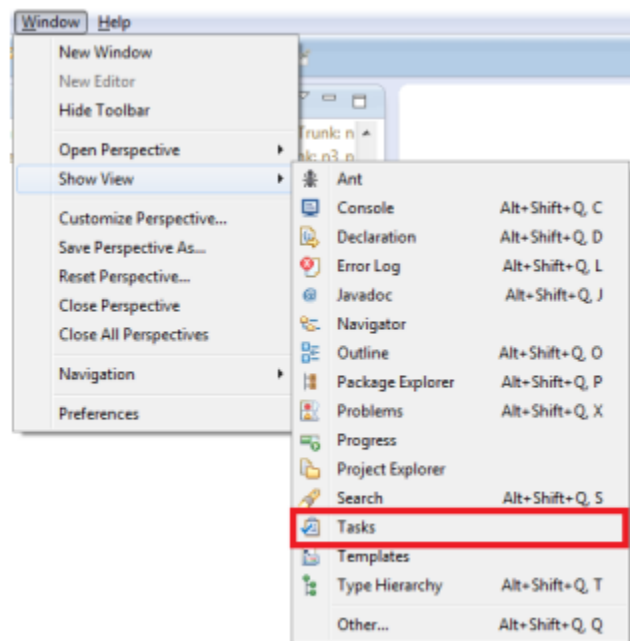
# INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

- “Abrir implementaciones en Eclipse”. Este video le explicará cómo dirigirse a la implementación (declaración) de una clase, método y atributo (objeto).
6. Desde Eclipse revise la documentación de las clases del mundo. Esto le permitirá entender para qué sirve cada método y cada uno de los atributos. El video “Lectura de contratos de un método”, disponible en el enlace indicado en la página “Videos de ayuda” de la sección “Recursos”, explica cómo leer la documentación de un método.

## ETAPA 3: IMPLEMENTACIÓN

En esta etapa usted debe crear una clase, agregar atributos y completar métodos faltantes en el mundo del problema de forma que el programa funcione correctamente.

Los comentarios **TODO** (*to do – por hacer*) le indicarán los puntos donde hay que incluir instrucciones. Puede apoyarse también en la documentación de los métodos disponible en el código. Para visualizar los TODOs vaya al menú Window > Show View > Tasks como se muestra en la siguiente figura:





## INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

Para facilitar el desarrollo de los TODOs, se recomienda ver el video “Manejo de TODOs en Eclipse”, disponible en el enlace indicado en la página “Videos de ayuda” de la sección “Recursos”.

Opcionalmente, se recomienda ver el video “Uso de renombrado seguro en Eclipse”, el cual le enseña a renombrar un método o un atributo de forma automática y segura en caso de haber cometido un error en su nombramiento. Este video se encuentra en el enlace indicado en la página “Videos de ayuda” de la sección “Recursos”.

Por último, el video “Abrir jerarquía de llamados en Eclipse” le enseña cómo determinar en dónde se está usando (se llama) un atributo o un método en un programa. Este video se encuentra en el enlace indicado en la página “Videos de ayuda” de la sección “Recursos”.

A continuación serán presentados los pasos sugeridos para completar el programa.

### Crear la clase LineaVozIP

En esta parte usted debe crear la clase **LineaVozIP** a partir de las siguientes indicaciones. Para este paso apóyese del modelo conceptual del ejercicio.

1. Cree la clase **LineaVozIP** la cual hereda de la clase **LineaTelefonica**.
2. Adicione y documente (con un comentario descriptivo) las constantes y los atributos mencionados en la descripción del problema, y que correspondan al modelo del mundo.
3. Adicione el método constructor de la clase. Este método no recibe parámetros y debe inicializar el saldo disponible en \$100000, y la cantidad de minutos a larga distancia y celular en cero.
4. Adicione y documente con un comentario descriptivo los métodos analizadores:
  - darNumeroMinutosLargaDistancia
  - darNumeroMinutosCelular
  - darSaldoDisponible



## INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

5. Adicione los métodos que modifican los atributos de la clase o de la clase de la cual hereda:
  - `agregarLlamadaCelular`
  - `agregarLlamadaLargaDistancia`

### Validar la clase `LineaVozIP`

Compruebe la validez del código implementado en la clase **`LineaVozIP`**. Para esto ejecute la prueba unitaria **`LineaVozIPTest.java`** que se encuentra en el paquete **`uniandes.cupi2.lineasTelefonicas.test`**

Recuerde que en el menú principal del curso, localizado en la parte izquierda de la ventana, encontrará la sección “Recursos”. Diríjase a “Videos de ayuda”, siga el enlace allí indicado y vea el video “Manejo de pruebas en Eclipse”, el cual le enseña cómo ejecutar las pruebas unitarias.

### Completar la clase `Empresa`

En esta parte usted debe completar la clase **`Empresa`**. Para esto, siga las siguientes indicaciones:

1. Agregue la asociación “`lineaVozIP1`” que modela una línea VozIP.
2. Complete los siguientes métodos:
  - `Empresa`
  - `darLineaVozIP1`
  - `agregarLlamadaLocalLineaVozIP1`
  - `agregarLlamadaLargaDistanciaLineaVozIP1`
  - `agregarLlamadaCelularLineaVozIP1`
  - `darTotalNumeroLlamadasDesdeLineasAlternativas`
  - `darTotalMinutosDesdeLineasAlternativas`
  - `darTotalCostoLlamadasDesdeLineasAlternativas`

### Validar la clase `Empresa`

Compruebe la validez del código implementado en la clase **`Empresa`**. Para esto ejecute la prueba unitaria **`EmpresaTest.java`** que se encuentra en el paquete **`uniandes.cupi2.lineasTelefonicas.test`**



## INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

Recuerde que en el menú principal del curso, localizado en la parte izquierda de la ventana, encontrará la sección “Recursos”. Diríjase a “Videos de ayuda”, siga el enlace allí indicado y vea el video “Manejo de pruebas en Eclipse”, el cual le enseña cómo ejecutar las pruebas unitarias.

### ETAPA 4: INTERACCIÓN CON EL PROGRAMA

Ejecute el programa e interactúe con todas las opciones disponibles en la interfaz para verificar la funcionalidad del programa.