Actividad autocorregible: Ingeniería de *software* dirigida por modelos

**Descripción de la actividad**

Recuerda que esta actividad deberá ser completada en la plataforma. Solo tendrás un intento. Una vez que la envíes, se dará por cerrada. Tu puntuación y las respuestas correctas se mostrarán cuando finalice el periodo de entrega de la tarea.

En esta actividad vamos a profundizar en la ingeniería dirigida por modelos creando una serie de lenguajes específicos de dominio. Estos los definiremos de forma gráfica mediante metamodelos y haciendo uso de la herramienta Eclipse Modeling Framework (EMF).

Por ello, lo primero será instalar EMF. Para hacerlo iremos a Eclipse Foundation (<https://www.eclipse.org/downloads/>) y descargaremos el Eclipse Installer en nuestro equipo.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Figura 1. Captura de la página de descargas de eclipse. Fuente: elaboración propia.

Iniciamos el instalador:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 2. Captura del instalador. Fuente: elaboración propia.

Bajaremos hasta seleccionar Eclipse Modeling Tools:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 3. Captura del instalador. Selección EFM. Fuente: elaboración propia.

Seleccionaremos una carpeta de destino —se recomienda no ubicarlo en rutas que tengan espacios en blanco como Archivos de Programa; mejor en C:\tools o en C:\tools\emf—:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 4. Captura del instalador. Selección de carpeta de destino. Fuente: elaboración propia.

Aceptaremos los acuerdos de licencia:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 5. Captura del instalador. Licencia. Fuente: elaboración propia.

El instalador comenzará:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Figura 6. Captura del instalador. Instalación en curso. Fuente: elaboración propia.

En función de la conectividad, es posible que muestre mensajes de advertencia o incluso que sea necesario repetir el proceso de instalación:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 7. Captura del instalador. Mensajes de aviso. Fuente: elaboración propia.

Una vez que ha finalizado la instalación podemos lanzar el EFM desde el propio instalador, o bien ejecutando el eclipse.exe que tendremos en la carpeta destino:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 8. Captura del instalador. Mensaje final. Fuente: elaboración propia.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Figura 9. Carpeta con la instalación y aplicación eclipse.exe. Fuente: elaboración propia.

Como lo primero que nos va a pedir es un espacio de trabajo donde almacenar los proyectos, es recomendable tener una carpeta *workspace* creada.

* Inicializamos eclipse:

Patrón de fondo

Descripción generada automáticamente

Figura 10. Captura de eclipse. Inicio. Fuente: elaboración propia.

* Seleccionamos una carpeta como espacio de trabajo:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 11. Captura de eclipse. Workspace. Fuente: elaboración propia.

* Tras unos segundos tendremos el entorno disponible:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 12. Captura de eclipse. Home. Fuente: elaboración propia.

* Pulsaremos arriba a la derecha para ocultar (Hide) y tendremos la vista vacía:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 13. Captura de eclipse. IDE vacío. Fuente: elaboración propia.

1. Vamos a empezar a familiarizarnos con el entorno EMF.

Crea tu primer EMF Modeling Project:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ponle como nombre «ListaCompra» y pulsa Finish:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Veremos que se ha creado el proyecto y tenemos esta visualización para poder crear metamodelos en base a ECore, la cual es una especificación similar a Essential MoF:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En el menú de la derecha haz clic sobre Class y pulsa en el canvas blanco del medio para crear una nueva clase:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Denomínala «ListaCompra»:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Haz lo mismo con otra que se llame «Producto»:

Diagrama

Descripción generada automáticamente

Pulsa en la derecha en Attribute y añade un nombre y una cantidad a Producto:

Interfaz de usuario gráfica

Descripción generada automáticamente

En la paleta de propiedades, seleccionaremos el EType como un string (EString):

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Tras añadir el nombre nos quedará así:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Añadiremos la cantidad:

Interfaz de usuario gráfica

Descripción generada automáticamente

Ya solo nos falta relacionar la lista con el producto. Para esto, seleccionaremos Composition, que está bajo Relation, pulsaremos primero en ListaCompra y posteriormente en Producto:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Y renombramos la relación a productos:

Imagen que contiene Gráfico en cascada

Descripción generada automáticamente

Ya tenemos nuestro primer metamodelo ECore desarrollado en el EMF, el listaCompra.ecore:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Sobre el canvas, con el botón derecho del ratón pulsa Generate y luego All:

Diagrama

Descripción generada automáticamente

¿Qué es lo que ha ocurrido?

* 1. Nada.
  2. Muestra en consola un mensaje con un error.
  3. Ha creado código en mi proyecto en la carpeta src.
  4. Ha creado código en mi proyecto en la carpeta src y, además, ha creado otros dos proyectos nuevos.

1. Ya has creado tu primer metamodelo para simplemente guardar la lista de la compra. Vamos a dar un paso más para crear nuestro primer modelo, que es conforme al metamodelo anterior. Para ello, vamos a utilizar el proyecto «.editor» que en el paso anterior el EMF generó.

Pulsa sobre el proyecto «.editor» con el botón derecho del ratón. En el menú contextual, selecciona Run AS y Eclipse Application:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al ser un *plugin* de eclipse podemos ejecutarlo dentro del propio programa. Para no tener que generar una versión e instalarla, podemos, desde una instancia de eclipse (en la que estábamos editando el metamodelo), lanzar otra segunda instancia que ejecutará ese *plugin*. Saldrá de nuevo que está arrancado el IDE. Esta vez no pedirá el *workspace*, sino que guardará todo por defecto bajo una carpeta runtime-EclipseApplication (por ejemplo, C:\tools\EMF\eclipse\runtime-EclipseApplication si lo instalaste en C:\tools\EMF):

Patrón de fondo

Descripción generada automáticamente

Al ser la primera ejecución estará vacío, aunque en el título de la aplicación veremos que pone runtime-EclipseApplication:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Entraremos en New > Modeling Project:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Le pondremos un nombre —por ejemplo, «EjemploListaCompra»— y pulsaremos Finish:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Nos creará un proyecto vacío:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Sobre el proyecto, entraremos en New > Other:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En el buscador pondremos Lista y seleccionaremos el *wizard* siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Le pondremos un nombre —por ejemplo, «Supermercado1»—:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Le indicaremos el elemento raíz del modelo (en nuestro caso, la lista de la compra):

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Tras pulsar Finish, ¿qué ha ocurrido?

* 1. Ha generado código sobre el proyecto bajo src.
  2. Ha generado el modelo y nos muestra una visualización en modo Selection-Parent-List-Tree-Table-Tree with Columns.
  3. Ha generado el modelo y nos muestra una visualización en modo Tree-Table-Tree with Columns.
  4. Genera un diagrama visual con un modelo de lista de compra de ejemplo.

1. Ya solo nos queda un paso: incluir datos en el modelo.

Dentro de la lista de la compra, con el botón derecho hay que seleccionar New Child y Producto:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En el editor inferior pondremos el nombre; por ejemplo, «Leche»:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Hacemos este ejercicio varias veces modificando el nombre y la cantidad:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Intenta editar la cantidad de zumo y pon «un par de litros»:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

¿Qué ocurre?

* 1. Nada.
  2. Salta una excepción y se cierra eclipse.
  3. Muestra un mensaje de «NumberFormatException» y no deja asignar valor.
  4. Deja el valor asociado correctamente.

1. En este metamodelo solo podemos tener una lista de la compra y, además, no podemos organizarla por zonas o supermercados. ¿Cómo extenderías el metamodelo para cumplir las siguientes extensiones?
   * Una lista de compra tiene sublistas.
   * Incluyen una descripción del lugar.
   * Cada una tiene productos como hasta ahora.
   1. Se debe incluir un nuevo elemento al metamodelo sublista.
   2. Se debe incluir un nuevo atributo a la lista y una composición reflexiva.
   3. En metamodelos no se pueden anidar elementos de forma recursiva.
   4. El modelo existente ya lo soportaba.
2. Vamos a crear el primer metamodelo propio que nos permita definir páginas web simplificadas.

Crea el ecore que soporta la definición de una página web con zonas (header, central y footer) y elementos (label, buton, image y link) como para poder generar este modelo:

Texto

Descripción generada automáticamente con confianza media

¿Qué contiene tu metamodelo?

* 1. 10 clases (2 de ellas abstractas) 7 herencias y 2 composiciones.
  2. 8 clases (0 abstractas), 3 herencias y 2 composiciones.
  3. 3 clases y 2 composiciones.
  4. No se puede generar el metamodelo para cumplir esa especificación.

1. Crea un nuevo metamodelo para soportar un *workflow* de pasos simple, que permita registrar una secuencia de actividades secuencial unida por transiciones para soportar un modelo de este estilo:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

¿Qué contiene tu metamodelo?

* 1. 3 clases y 1 composición.
  2. 4 clases (1 abstracta) 1 composición, 2 herencias y 2 relaciones.
  3. 4 clases (0 abstractas) y 3 composiciones.
  4. No se puede generar el metamodelo para cumplir esa especificación.

1. Queremos generar un metamodelo de bases de datos. Vamos a comenzar definiendo las tablas y sus columnas, de momento sin relaciones, para poder cubrir un modelo de este tipo:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

¿Qué contiene tu metamodelo?

* 1. 2 clases y 1 composición.
  2. 4 clases (0 abstractas) y 3 composiciones.
  3. No se puede generar el metamodelo para cumplir esa especificación.
  4. 4 clases (1 abstracta), 2 composiciones y un tipo enumerado.

1. Extendemos el metamodelo de esta forma:

Diagrama

Descripción generada automáticamente

https://github.ibm.com/APCC-Cloud/apcc-ibmcarbon-vuejs/pull/386

¿Es posible representar un modelo así?

Texto

Descripción generada automáticamente con confianza media

Donde en el préstamo tenemos:

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

* 1. No, ya que Foreign Key tendría que ser otra entidad separada de la columna.
  2. No, ya que no puede ser Foreign Key y clave primaria a la vez.
  3. Sí, el metamodelo lo soporta.
  4. No, porque, aunque el metamodelo lo soportara, el modelo sería erróneo.

1. ¿Cómo extenderías el metamodelo para incluir la restricción de valor único para una columna (de momento simplemente asociado a un campo)?
   1. Hay que añadir un nuevo atributo *boolean* en Tabla.
   2. Hay que añadir un nuevo elemento como ForeignKey que herede de Columna.
   3. Hay que añadir un nuevo atributo *boolean* en Columna.
   4. Hay que añadir un nuevo atributo *boolean* en Table.
2. ¿Qué tendríamos que modificar para soportar el modelado de que no podamos tener una persona que tiene el mismo nombre y teléfono, pero sí que tengamos dos personas con el mismo nombre o con el mismo teléfono?
   1. Crear un nuevo elemento que se relacione con Column.
   2. Crear un nuevo elemento que herede de Column.
   3. Crear un nuevo elemento que herede de Foreign Key.
   4. No es posible incluirlo en el metamodelo anterior.