	Universidad Tecnológica Nacional
	Facultad Regional Buenos Aires
	Análisis de Sistemas de Información 2024
	Curso: K2001 Turno: Mañana

**Nombre del Ejercicio**  
**Herramientas CASE**  
**Tema**  
**Sistemas – Sistemas de Información**

GRUPO N° 2	
Nombre y Apellido	Legajo
Agustín Nicolás Herzkovich	213.787-2
Ezequiel Su	214.167-0
Facundo Martín Gutman	212.989-9
Facundo Tobías Sabelli	214.095-0
Fausto Oliva	213.943-1
Luciano Ezequiel Tapia	214.176-0
Tomás Pedro Palazzesi	213.961-3

<b>FECHA DE PRESENTACIÓN:</b>	26/08/2024
<b>FECHA DE DEVOLUCIÓN:</b>	
<b>CALIFICACIÓN:</b>	
<b>FIRMA PROFESOR:</b>	

Análisis de Sistemas de Información		
Tema: Sistemas – Sistemas de Información	Nombre del Ejercicio: Herramientas CASE	Grupo: 2

## Tabla de Contenidos

Introducción .....	2
Objetivo del informe.....	2
Importancia de las herramientas CASE en el desarrollo de sistemas. ....	2
Concepto de Herramienta CASE.....	2
Definición de herramienta CASE.....	2
Clasificación general de Herramientas CASE.....	2
• Herramientas I-CASE .....	2
• Herramientas Upper CASE .....	2
• Herramientas Lower CASE.....	2
Breve historia y evolución de las herramientas CASE .....	3
Relación entre Herramienta CASE y Metodología de Sistemas.....	3
Vinculación con la metodología de sistemas .....	3
Caso particular en el análisis de sistemas .....	4
Características que debe cumplir una Herramienta CASE.....	4
Listado de características esenciales .....	4
Listado de características deseables.....	4
Evaluación Comparativa de dos Herramientas CASE.....	5
Presentación de las herramientas seleccionadas.....	5
Enterprise Architect.....	5
Rational Rose .....	5
Tabla comparativa .....	6
Demo de Enterprise Architect .....	7
Presentación de la demo .....	7
Pantalla de inicio .....	7
Modelado UML.....	7
Generación de código .....	9
Simulación y validación .....	10
Video .....	10
Referencias Bibliográficas .....	10

Análisis de Sistemas de Información		
Tema: Sistemas – Sistemas de Información	Nombre del Ejercicio: Herramientas CASE	Grupo: 2

## **Introducción**

### **Objetivo del informe**

El objetivo de este informe es analizar el uso de las herramientas CASE en el desarrollo de sistemas. A lo largo del trabajo, se explicará qué son las herramientas CASE, cómo se relacionan con las metodologías de desarrollo de sistemas, y se destacarán las características más importantes que deben tener. También se realizará una comparación entre dos herramientas CASE para ver sus diferencias, ventajas y desventajas. Por último, se incluirá una demostración de una de las herramientas, mostrando cómo se aplica en la práctica dentro de un proyecto de software.

### **Importancia de las herramientas CASE en el desarrollo de sistemas.**

Las herramientas CASE tienen un papel fundamental en el desarrollo de sistemas, favoreciendo la automatización y mejora de distintas fases del ciclo de vida del software. Estas herramientas no solo aumentan la productividad al reducir el tiempo utilizado en tareas repetitivas, sino que también proporcionan un entorno estructurado que mantiene estándares de calidad consistentes a lo largo del desarrollo. Además, promueven una colaboración más efectiva entre los miembros del equipo al ofrecer una vista centralizada y coherente del proceso de desarrollo [1][2].

## **Concepto de Herramienta CASE**

### **Definición de herramienta CASE**

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo en términos de tiempo y dinero. Al utilizar herramientas CASE, cambia radicalmente la manera en la que el código y la documentación de los programas son modificados y actualizados. El mayor beneficio es que el mantenimiento de los sistemas es realizado en el nivel más alto de la abstracción (análisis-diseño) y no en el nivel más bajo (código-fuente). [3]

### **Clasificación general de Herramientas CASE**

Existen diferentes tipos de herramientas CASE: integradas (I-CASE), front-end (Upper CASE) y back-end (Lower CASE).

- **Herramientas I-CASE**

Las herramientas I-CASE abarcan todas las fases del ciclo de vida del desarrollo y permiten la integración y el intercambio de datos entre herramientas mediante un repositorio compartido. Este repositorio centraliza la información y está disponible para todas las herramientas durante todo el ciclo de vida del proyecto.

- **Herramientas Upper CASE**

Las herramientas Upper CASE se utilizan en las primeras etapas de la metodología de sistemas, como la planificación del proyecto, el análisis de sistemas y el diseño general. Ayudan a diagramar y definir tanto el sistema actual como el propuesto.

- **Herramientas Lower CASE**

Las herramientas Lower CASE se emplean en las etapas finales de la metodología de sistemas, incluyendo el diseño detallado del sistema, la implementación y el soporte. Estas herramientas comúnmente incluyen generadores de código, compiladores y herramientas de soporte para pruebas.

[4]

Análisis de Sistemas de Información		
Tema: Sistemas – Sistemas de Información	Nombre del Ejercicio: Herramientas CASE	Grupo: 2

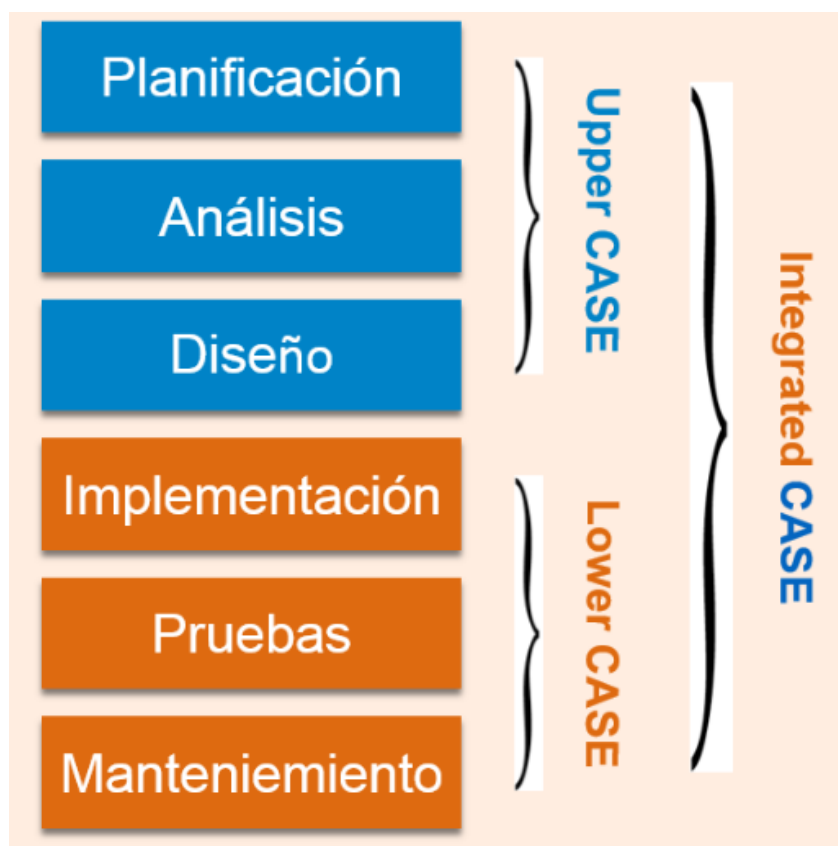


Figura 1: Tipos de Herramientas CASE [5].

#### Breve historia y evolución de las herramientas CASE

Las herramientas de software CASE surgieron en la década de 1970 como una categoría de productos significativa, desarrolladas para aportar orden y disciplina a grandes proyectos de desarrollo de software. Los proveedores de herramientas CASE afirmaban que estos productos mejorarían la productividad en TI y reducirían los errores en el desarrollo de software.

Con el tiempo, las herramientas CASE han evolucionado considerablemente. Las primeras herramientas se centraban en automatizar algunos procesos, como la documentación. Posteriormente, las herramientas incorporaron características para el diseño de software, generación de código, modelado de procesos y control de versiones. Además, empezaron a incluir integraciones incorporadas para trabajar de manera fluida con otras herramientas de desarrollo de software.

Hoy en día, las herramientas CASE integran tecnologías modernas como inteligencia artificial y computación en la nube, y también son compatibles con programación visual, programación orientada a objetos y procesos de desarrollo ágil. [6]

### **Relación entre Herramienta CASE y Metodología de Sistemas**

#### Vinculación con la metodología de sistemas

La metodología de sistemas define una serie de pasos a seguir (etapas) para construir un artefacto cuyo buen uso soluciona un problema en el campo de los sistemas de información. Las herramientas CASE brindan soporte técnico en cada etapa, desde la captura de requisitos hasta el mantenimiento del sistema.

Por ejemplo, un enfoque en la fase de análisis puede requerir la creación de un modelo que describa los procesos de negocio o la estructura de datos del sistema. Las herramientas CASE facilitan esta tarea al generar automáticamente diagramas y modelos, garantizando coherencia e integridad.

Análisis de Sistemas de Información		
Tema: Sistemas – Sistemas de Información	Nombre del Ejercicio: Herramientas CASE	Grupo: 2

Existe una relación **sinérgica** entre las herramientas CASE y la metodología de sistemas. Si bien la metodología proporciona la estructura necesaria para el desarrollo de sistemas, las herramientas CASE proporcionan mecanismos para aplicar estos principios de manera eficaz y eficiente, garantizando coherencia y calidad durante todo el ciclo de vida del desarrollo de software.

#### Caso particular en el análisis de sistemas

Las herramientas CASE son de utilidad en cada una de las etapas del análisis de sistemas. Dentro de este marco, estas ofrecen funcionalidades que permiten automatizar y mejorar la precisión de las actividades que se realizan en cada etapa del análisis.

1. **Reconocimiento:** En esta fase, donde se identifican los problemas y oportunidades en el sistema actual, las herramientas CASE permiten documentar de manera eficiente las observaciones iniciales. Facilitan la comprensión de los problemas existentes en la organización.
2. **Relevamiento:** Las herramientas CASE pueden ayudar a estructurar la información recopilada mediante diagramas de flujo, modelos de datos y otros artefactos visuales que permiten una visión más clara del sistema existente.
3. **Diagnóstico:** La capacidad de modelar diferentes escenarios y alternativas que brinda estas herramientas facilita la visualización de los posibles impactos de los cambios propuestos en el sistema.
4. **Estudio de Factibilidad:** Las herramientas CASE también juegan un papel crucial en el estudio de factibilidad, permitiendo realizar simulaciones y análisis de costos y beneficios, y ayudando a determinar si las soluciones propuestas son viables técnica, económica y operativamente.
5. **Análisis de Requisitos:** Permiten especificar y validar los requisitos del sistema de manera estructurada. Facilitan la creación de modelos detallados que describen cómo debería funcionar el sistema, asegurando que todos los requisitos se capturen y validen correctamente antes de pasar a la fase de diseño.

### Características que debe cumplir una Herramienta CASE

#### Listado de características esenciales

- ❖ **Entorno de desarrollo interactivo:** Permite trabajar de forma interactiva sobre un proyecto creando nuevos elementos y viendo los resultados en el acto. El cual debe ser intuitivo y accesible para usuarios con diferentes niveles de experiencia.
- ❖ **Tiempo de respuesta rápido:** Cualquier modificación, o elemento que añadamos producirá una alteración inmediata en el sistema.
- ❖ **Comprobación de errores:** El programa comprueba directamente la coherencia, consistencia e integridad del sistema.
- ❖ **Automatización de tareas de desarrollo y mantenimiento:** Las tareas repetitivas y rutinarias se realizan de forma automática mediante las especificaciones dadas por el usuario. Permitiendo así adaptar estos procesos a automáticos según las necesidades específicas del proyecto.

[7]

#### Listado de características deseables

- ❖ **Proporcionar topologías de aplicación flexibles:** La capacidad de separar la aplicación entre diferentes capas (cliente, servidor, y entre servidores) es deseable para aplicaciones complejas y distribuidas, mejorando la escalabilidad y el rendimiento.
- ❖ **Proporcionar aplicaciones portátiles:** La capacidad de generar código que sea portable a múltiples plataformas (Windows, OS/2, Macintosh, Unix) es altamente deseable en entornos heterogéneos y cuando se trabaja con diversas plataformas de servidor. La misma también hace referencia a la capacidad de integrar diferentes infraestructuras y servicios en la nube.
- ❖ **Control de Versión:** La herramienta debe reconocer las versiones de códigos que se ejecutan en los clientes y servidores, y asegurarse que sean consistentes. También, la herramienta debe ser capaz de controlar un gran número de tipos de objetos incluyendo texto, gráficos, mapas de bits, documentos complejos y objetos únicos, tales como definiciones de pantallas y de informes, archivos de objetos y datos de prueba y resultados. Debe

mantener versiones de objetos con niveles arbitrarios de granularidad; por ejemplo, una única definición de datos o una agrupación de módulos.

[8]

## **Evaluación Comparativa de dos Herramientas CASE**

### **Presentación de las herramientas seleccionadas**

#### **Enterprise Architect**

Descripción: Enterprise Architect es una herramienta CASE desarrollada por Sparx Systems. Es conocida por su capacidad para soportar la modelización orientada a objetos a través de UML. La herramienta también ofrece funciones para la gestión de requisitos, la simulación de procesos de negocio, y la ingeniería inversa.

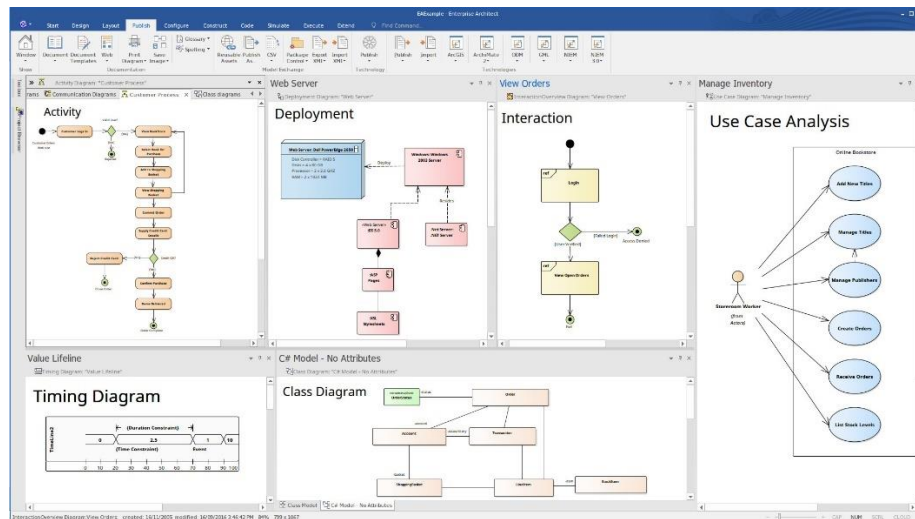


Figura 2: Modelado UML en Enterprise Architect [9].

#### **Ventajas:**

- Soporte extensivo para UML.
- Buenas capacidades de gestión de requisitos.
- Integración con otras herramientas de desarrollo y control de versiones.

#### **Desventajas:**

- Puede ser compleja de utilizar para principiantes.
- Requiere una curva de aprendizaje significativa.

#### **Ejemplo de uso:**

Un equipo de desarrollo de software utiliza Enterprise Architect para modelar los requisitos de un sistema de gestión empresarial, creando diagramas de clases y secuencias que permiten una comprensión clara y detallada del sistema antes de iniciar la codificación.

#### **Rational Rose**

Descripción: Rational Rose, desarrollado por IBM, es una de las herramientas CASE más antiguas y ampliamente utilizadas. También soporta modelado orientado a objetos y es compatible con varias metodologías de desarrollo de software.

## Análisis de Sistemas de Información

Tema: Sistemas – Sistemas de Información

Nombre del Ejercicio: Herramientas CASE

Grupo: 2

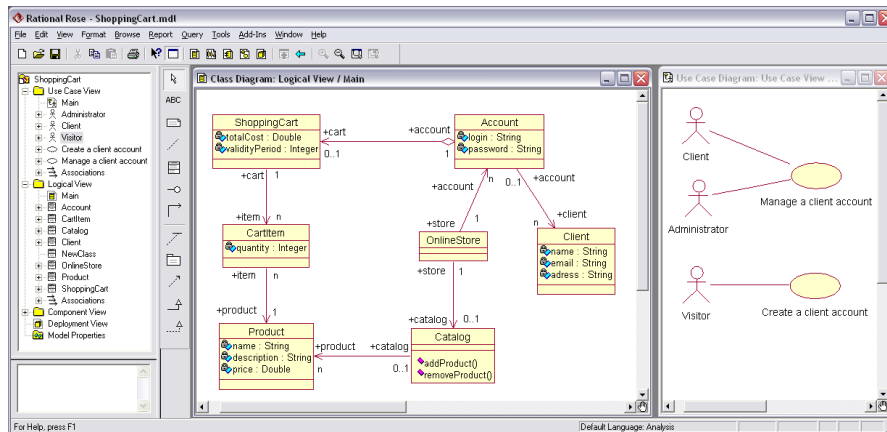


Figura 3: Modelado UML en Rational Rose [10].

Ventajas:

- Amplia base de usuarios y soporte técnico.
- Buena integración con otras herramientas de IBM y plataformas de desarrollo.

Desventajas:

- Interfaz de usuario algo obsoleta en comparación con herramientas más modernas.
- Limitada flexibilidad en comparación con competidores más recientes.

Ejemplo de uso:

Una empresa de desarrollo de software utiliza Rational Rose para modelar un sistema de inventario, utilizando diagramas de clases para definir las relaciones entre los diferentes componentes del sistema.

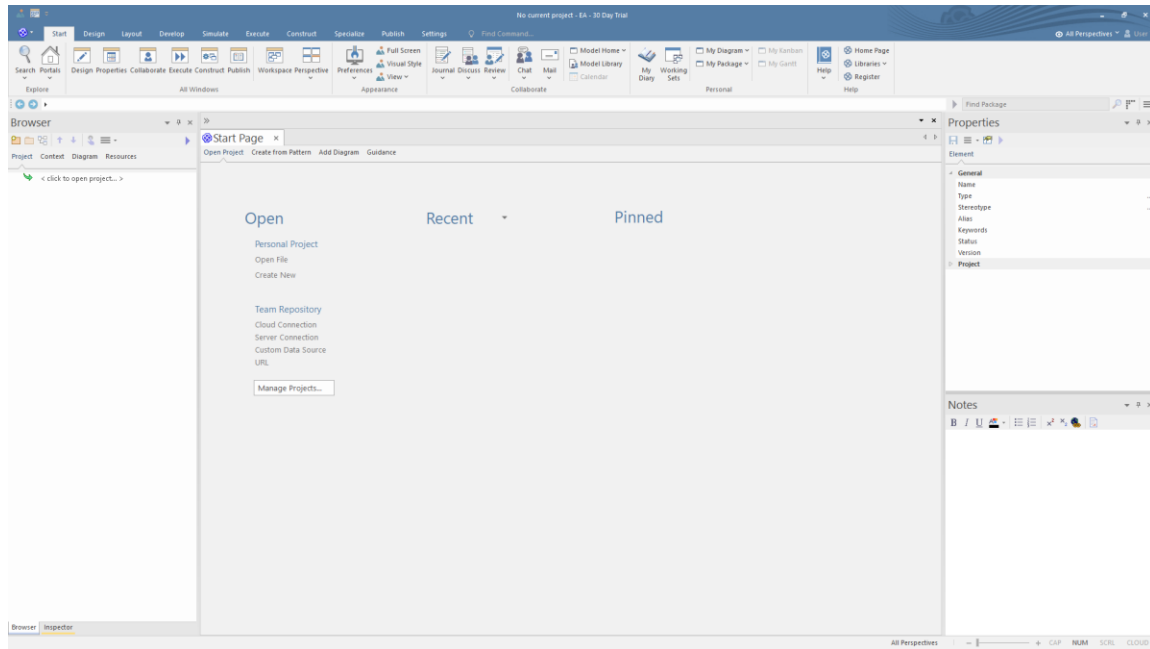
### Tabla comparativa

Características	Enterprise Architect	Rational Rose
Soporte para UML	Extensivo	Bueno
Gestión de Requisitos	Avanzado	Básico
Interfaz de Usuario	Moderna	Obsoleta
Integración con Otras Herramientas	Alta	Media
Facilidad de Uso	Media	Alta

## Demo de Enterprise Architect

### Presentación de la demo

#### Pantalla de inicio

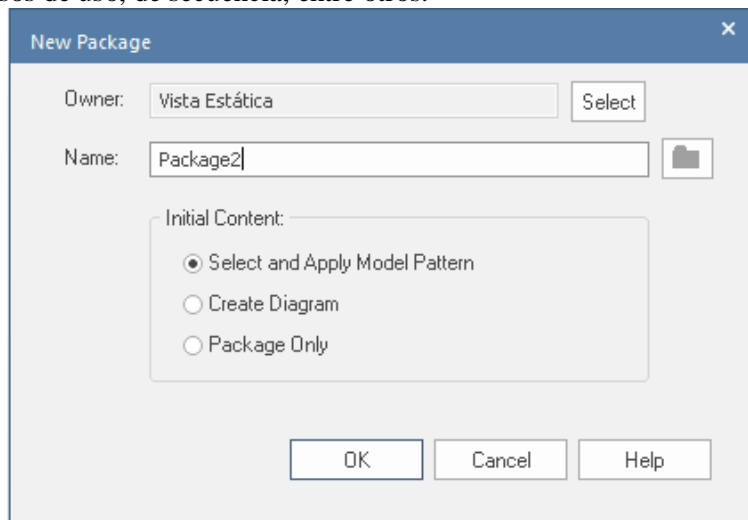


La interfaz inicial presenta un panel donde se puede abrir un proyecto existente o crear uno nuevo. Aquí, el usuario puede seleccionar plantillas predefinidas para trabajar en diferentes tipos de proyectos, como modelado de sistemas, diseño de bases de datos, o diagramas UML.

Además, la pantalla de inicio incluye accesos directos a las últimas noticias, tutoriales, y documentación, facilitando la orientación inicial para usuarios nuevos y experimentados.

#### Modelado UML

El modelado UML (Unified Modeling Language) es una de las funcionalidades más potentes de *Enterprise Architect*. Después de crear o abrir un proyecto, el usuario puede acceder a una variedad de diagramas UML, como diagramas de clases, de casos de uso, de secuencia, entre otros.





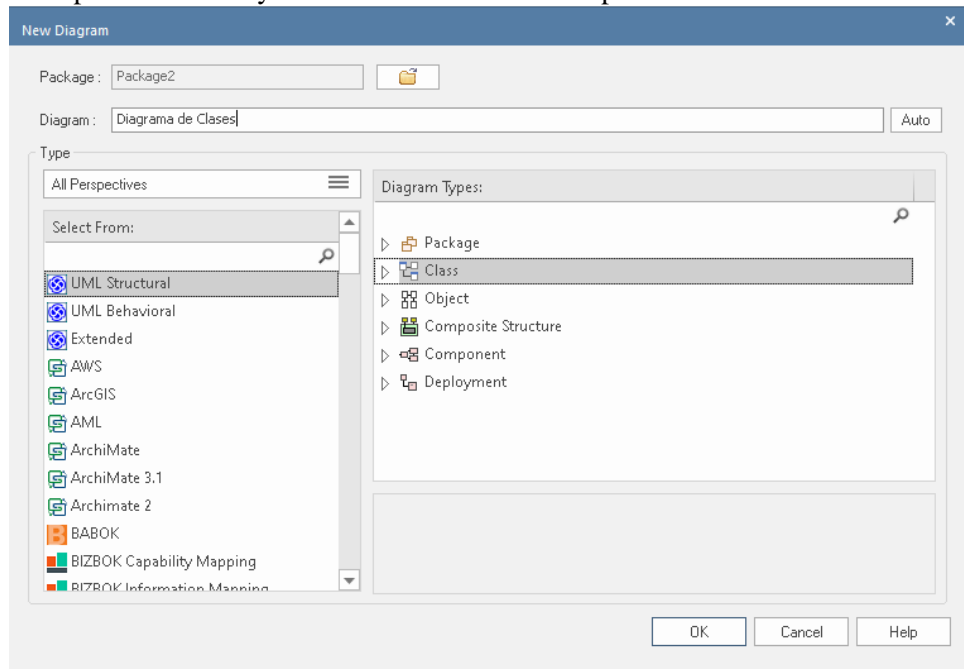
## Análisis de Sistemas de Información

Tema: Sistemas – Sistemas de Información

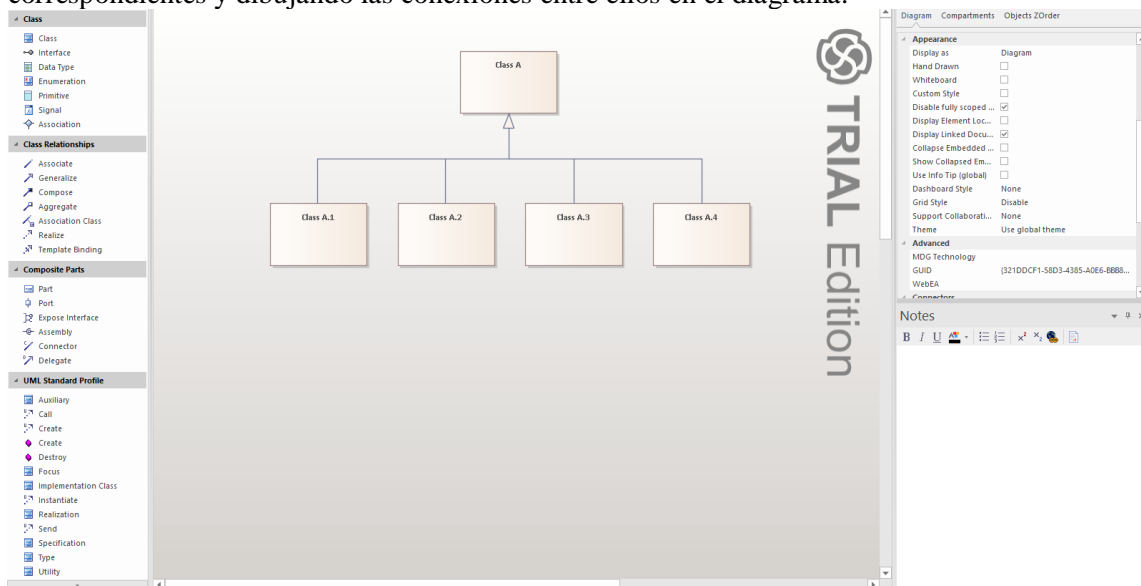
Nombre del Ejercicio: Herramientas CASE

Grupo: 2

Para crear un diagrama de clases, por ejemplo, el usuario selecciona la opción correspondiente y se abre un lienzo en blanco donde puede arrastrar y soltar elementos desde una paleta de herramientas.



Cada clase puede ser editada para agregar atributos, métodos, y establecer relaciones con otras clases mediante asociaciones, generalizaciones, o dependencias. Estas relaciones se crean simplemente seleccionando los elementos correspondientes y dibujando las conexiones entre ellos en el diagrama.



La interfaz de Enterprise Architect facilita la manipulación de estos modelos, permitiendo al usuario ajustar la disposición de los elementos, agregar notas explicativas, y realizar verificaciones de consistencia. Además, la herramienta ofrece la posibilidad de exportar estos diagramas en diversos formatos para su presentación o documentación.

En la Figura 4, se muestra el modelado de un Diagrama de Secuencia.

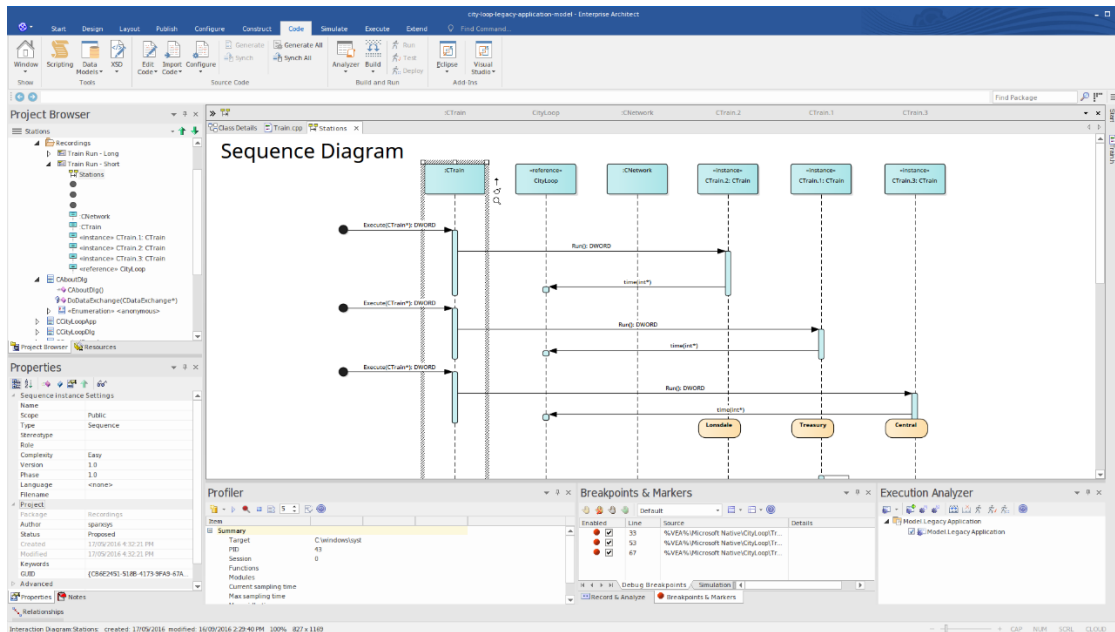


Figura 4: Automatización en Enterprise Architect [9].

## Generación de código

Una vez que el usuario ha terminado de modelar, puede seleccionar la opción de generación de código desde el menú de herramientas.

The 'Generate Code' dialog box is shown. It has a 'Path' field with a 'Browse' button. Below it, 'Target language' is set to 'Java' and 'Details' is set to 'Class A'. There are fields for 'Import(s) / Header(s)'. On the right side, there are buttons for 'Generate', 'Advanced', 'View', 'Save', 'Close', and 'Help'.

El usuario especifica el lenguaje de programación deseado (como Java, C++, C#, entre otros), y Enterprise Architect genera el cuerpo del código basado en las clases y relaciones definidas en el diagrama UML. Este código inicial puede incluir la estructura básica de las clases, con sus métodos y atributos, lo que ahorra tiempo y reduce el riesgo de errores manuales durante la codificación.



Análisis de Sistemas de Información		
Tema: Sistemas – Sistemas de Información	Nombre del Ejercicio: Herramientas CASE	Grupo: 2

- [5] “Software – CASE Herramientas,” tutorialspoint. Disponible en: [https://www.tutorialspoint.com/es/software\\_engineering/case\\_tools\\_overview.htm](https://www.tutorialspoint.com/es/software_engineering/case_tools_overview.htm). [Accedido: 17-Aug-2024].
- [6] R. Awati, “What is computer-aided software engineering (CASE)?,” TechTarget, 2024. [En línea]. Disponible: <https://www.techtarget.com/searcherp/definition/CASE-computer-aided-software-engineering>. [Accedido: 17-Aug-2024].
- [7] E. L. Cortés, “Ayudas automatizadas para el desarrollo del “software” (herramientas CASE). Tipos. Estructura. Prestaciones,” digital. [https://nube.burningthetowers.com/oposiciones/Temario/MAD/51.%20Ayudas%20automatizadas%20para%20el%20desarrollo%20de%20software%20\(herramientas%20CASE\).%20Tipos.pdf](https://nube.burningthetowers.com/oposiciones/Temario/MAD/51.%20Ayudas%20automatizadas%20para%20el%20desarrollo%20de%20software%20(herramientas%20CASE).%20Tipos.pdf).
- [8] M. Gaitan, “Herramientas CASE,” WordPress, 2012 [En línea]. Disponible: <https://marlonei.wordpress.com/herramientas-case/#:~:text=Características%20Deseables%20De%20Una%20Case&text=Ø%20Proporcionar%20topologías%20de%20aplicación,Ø%20Proporcionar%20aplicaciones%20portátiles>. [Accedido: 17-Aug-2024].
- [9] Sparx Systems, “Enterprise Architect”, [En Línea]. Disponible: <http://www.sparxsystems.com.ar/>. [Accedido: 17-Aug-2024].
- [10] Modelio, “Rational Rose to Modelio”, [En Línea]. Disponible: <https://www.modeliosoft.com/en/services/rational-rose-to-modelio.html>. [Accedido: 17-Aug-2024]
- [11] Sparx Systems, “Enterprise Architect”, [En Línea]. Disponible: <http://www.sparxsystems.com.ar/products/ea/> [Accedido: 17-Aug-2024].