



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ



FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

LICENCIATURA EN CIBERSEGURIDAD

PROGRAMACIÓN I

**INVESTIGACIÓN II
IDE NETBEANS VS ECLIPSE**

PREPARADO POR:

CABALLERO FÉLIX, 4-832-137

**A CONSIDERACIÓN DE
NAPOLEÓN IBARRA**

GRUPO: 2S311

II SEMESTRE

2025

PARTE I

ANÁLISIS

Los Integrated Development Environment IDE, constituyen herramientas fundamentales en la ingeniería de software, pues integran en una sola plataforma funciones como edición de código, compilación, depuración y gestión de proyectos. Entre los IDE más relevantes para el desarrollo en Java y otros lenguajes destacan Apache NetBeans y Eclipse, ambos de carácter libre y de código abierto.

NetBeans, actualmente bajo Apache Software Foundation, es un IDE multiplataforma que ofrece un conjunto de herramientas integradas listas para su uso, sin necesidad de configuraciones adicionales. Desde sus orígenes en 1996 y su posterior desarrollo por Sun Microsystems y Oracle, NetBeans se ha caracterizado por su facilidad de uso, interfaz intuitiva y soporte nativo para Java, PHP, HTML5 y C/C++ (Apache, 2024).

Eclipse, respaldado por la Eclipse Foundation y creado inicialmente por IBM en el año 2001, es uno de los IDE más difundidos en la industria del software. Se distingue por su arquitectura extensible basada en plugins y por un ecosistema sumamente amplio, que permite adaptarlo a diversos lenguajes y tecnologías, como Java, C/C++, Python, R y plataformas para la nube. Su robustez lo ha posicionado como una herramienta de referencia en entornos empresariales y proyectos a gran escala, aunque su curva de aprendizaje inicial es más pronunciada en comparación con NetBeans.



Fig 1 Logotipos oficiales de NetBeans y Eclipse

FORTALEZAS

NETBEANS: Destaca por ofrecer funcionalidades avanzadas listas para usar. Dispone de editores, asistentes (“wizards”) y plantillas para crear aplicaciones en Java, PHP, HTML5/CSS y otros lenguajes

Integra herramientas de refactorización, depuración y construcción (Maven, Gradle, etc.) sin requerir configuración inicial, lo que facilita el arranque inmediato de proyectos.

Su interfaz es relativamente intuitiva y estable: se considera «amigable para principiantes» y más consistente, al no depender tanto de plugins externos y también es multiplataforma Windows, Linux, macOS, BSD y sigue la filosofía “write once, run anywhere” de Java.

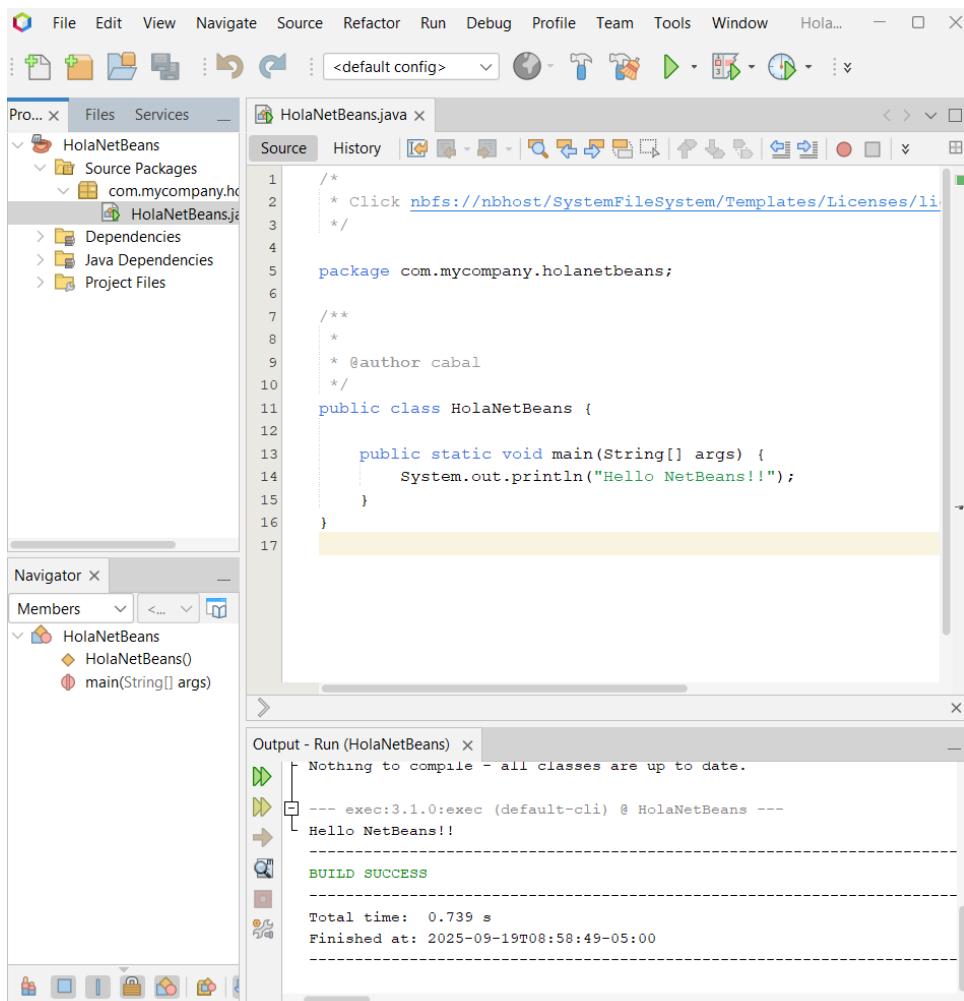


Fig 2 Interfaz del IDE NetBeans en ejecución.

ECLIPSE: Se caracteriza por su extensibilidad y flexibilidad. Cuenta con un enorme ecosistema de complementos: el Eclipse Marketplace ofrece alrededor de 1,500 plugins, frente a algo más de 1,100 en NetBeans

Eclipse incorpora sólidas herramientas de desarrollo empresarial: soporte nativo para Java (incluyendo la última versión Java 24), integración con control de versiones (EGit), gestión de proyectos (Maven, Gradle), edición de XML, depuración avanzada, entre otros, Asimismo, cuenta con una base de usuarios muy amplia y activa, respaldada por la Eclipse Foundation y varias empresas tecnológicas (IBM, Red Hat, Oracle, etc.), lo que favorece su evolución continua

Eclipse brinda una plataforma profesional robusta para proyectos grandes y heterogéneos: su arquitectura basada en plugins permite personalizar el entorno casi infinitamente.

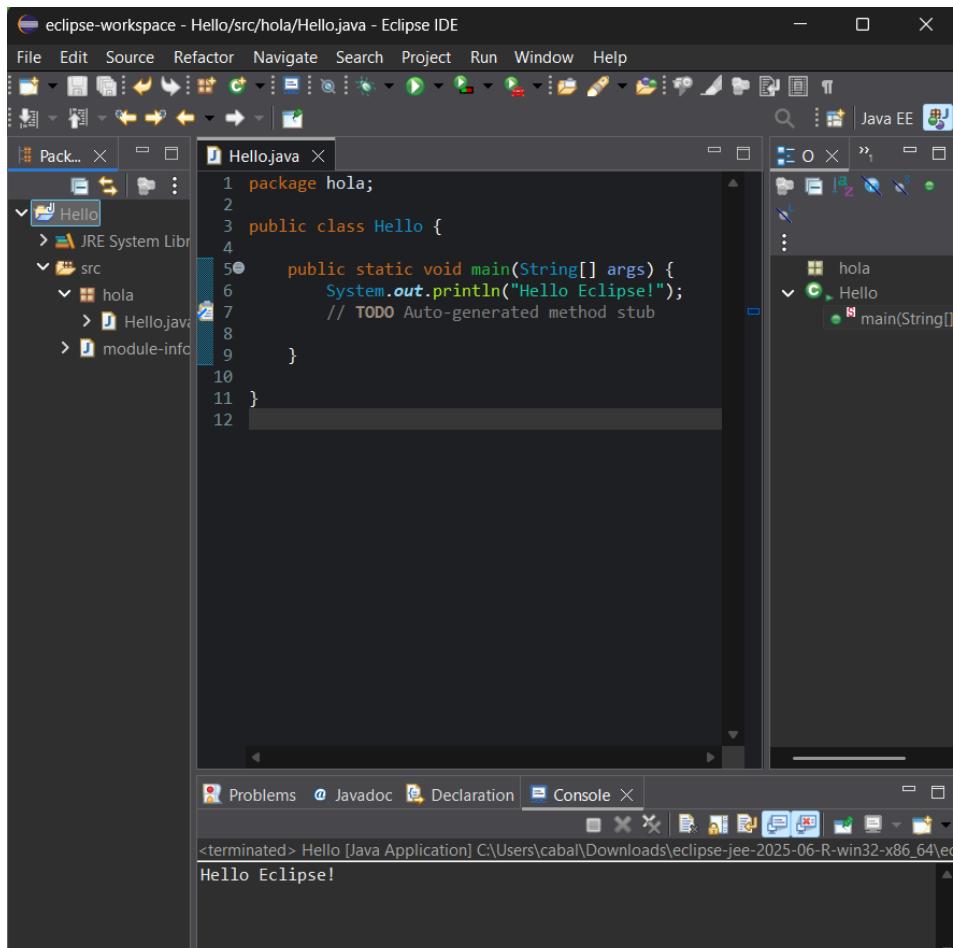


Fig 3 Interfaz del IDE Eclipse en ejecución.

CARACTERÍSTICA	NETBEANS	ECLIPSE
Facilidad de uso inicial	Entorno “listo para usar” con numerosas herramientas integradas con interfaz clara y coherente.	Tiene una curva de aprendizaje más pronunciada, interfaz básica que requiere añadir plugins. Muy flexible pero complejo al inicio.
Lenguajes soportados	Enfocado en Java, C/C++, PHP, HTML5/CSS y otros. Recientemente añade soporte inicial para Rust y se estudia Python.	Multilenguaje, Java, C/C++, C#, R, Python, etc. mediante extensiones. Soporte vía Eclipse Foundation.
Extensibilidad	Modular, pero ecosistema de plugins menor. Enfatiza funcionalidades integradas más que complementos externos.	Muy extensible, gran variedad de plugins y framework. Plataforma de código abierto bajo Eclipse Public License, que fomenta contribuciones externas.
Rendimiento y recursos	Puede consumir más memoria por sus componentes integrados y ser lento al indexar proyectos grandes	También es pesado en recursos, con interfaz más antigua. Permite manejar múltiples proyectos mediante instancias separadas o plugins.
Uso en la industria	Popular en proyectos independientes y entornos educativos. Soporta frameworks Java EE/Web, útil en desarrollo web (HTML5, PHP)	Ampliamente usado en el ámbito profesional y de grandes organizaciones. Base de usuarios consolidada en aplicaciones empresariales.

OPORTUNIDADES

ACADÉMICAS: Ambos IDE de código abierto son herramientas atractivas para la docencia de programación. Eclipse ofrece una edición “IDE for Education” adaptada a estudiantes universitarios, con soporte para Java y otros lenguajes educativos.

PROFESIONALES: En la industria del software, Eclipse sigue presente como plataforma de referencia para Java y proyectos de ingeniería. NetBeans continúa siendo útil en equipos pequeños o startups independientes, donde su estabilidad y conjunto integrado agiliza el desarrollo rápido.

DESARROLLO DE SOFTWARE: La demanda de multiplataforma y lenguajes emergentes abre nuevas oportunidades. Por ejemplo, Apache NetBeans ha incorporado soporte inicial para Rust y está trabajando en módulos de Python, lo que le permite seguir siendo relevante más allá de Java. Eclipse, gracias a su modelo de plugins. Ambos IDEs cuentan con integración nativa con sistemas de control de versiones y herramientas de construcción que son esenciales en entornos profesionales modernos.

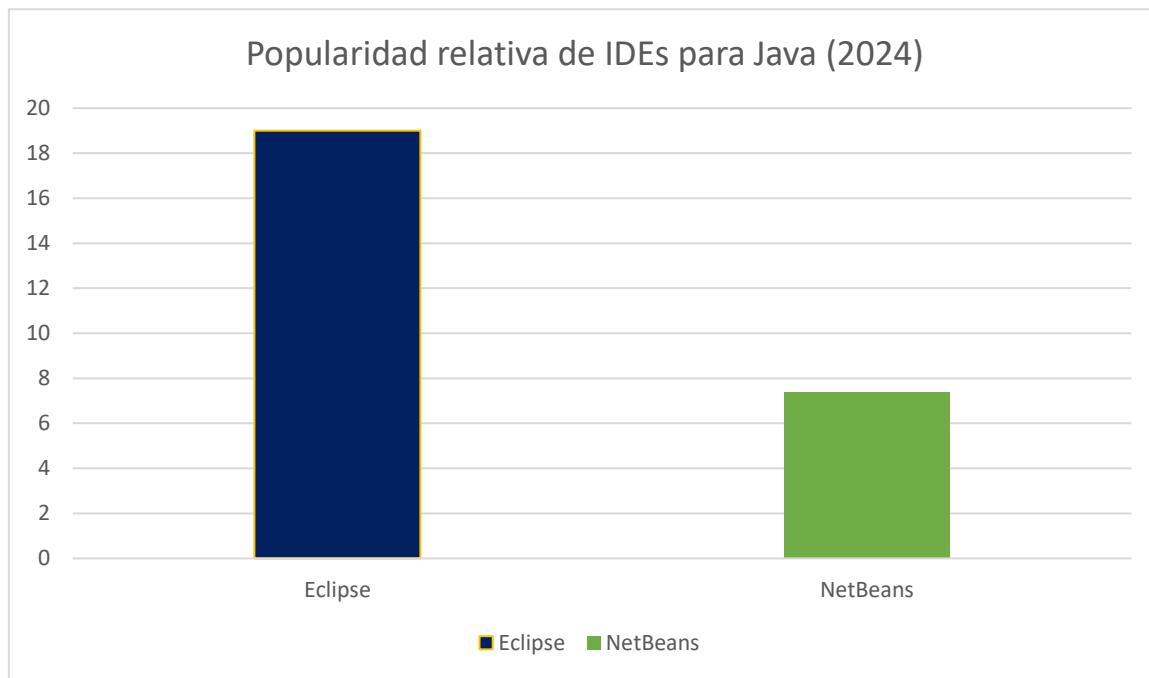


Fig 4 Los porcentajes se basan en la encuesta realizada por Baeldung en 2024 a 5,788 desarrolladores Java. Según los resultados, Eclipse fue utilizado por el 19 % de los encuestados y NetBeans por el 7.4 %.

DEBILIDADES

NETBEANS: Su principal limitación es el desempeño en equipos modestos o proyectos muy grandes, al incluir muchas herramientas integradas, puede necesitar más memoria y tiempo de carga que editores ligeros. Aunque estable, su ecosistema de extensiones es más reducido que el de Eclipse, por lo que carece de tantos plugins comunitarios, ciertas funcionalidades requieren extensiones, lo cual puede ralentizar el IDE cuando se cargan plugins adicionales. Algunos usuarios señalan que el código auto generado es malo y menos óptimo, lo que aumenta la dependencia del IDE y reduce el control sobre el código limpio.

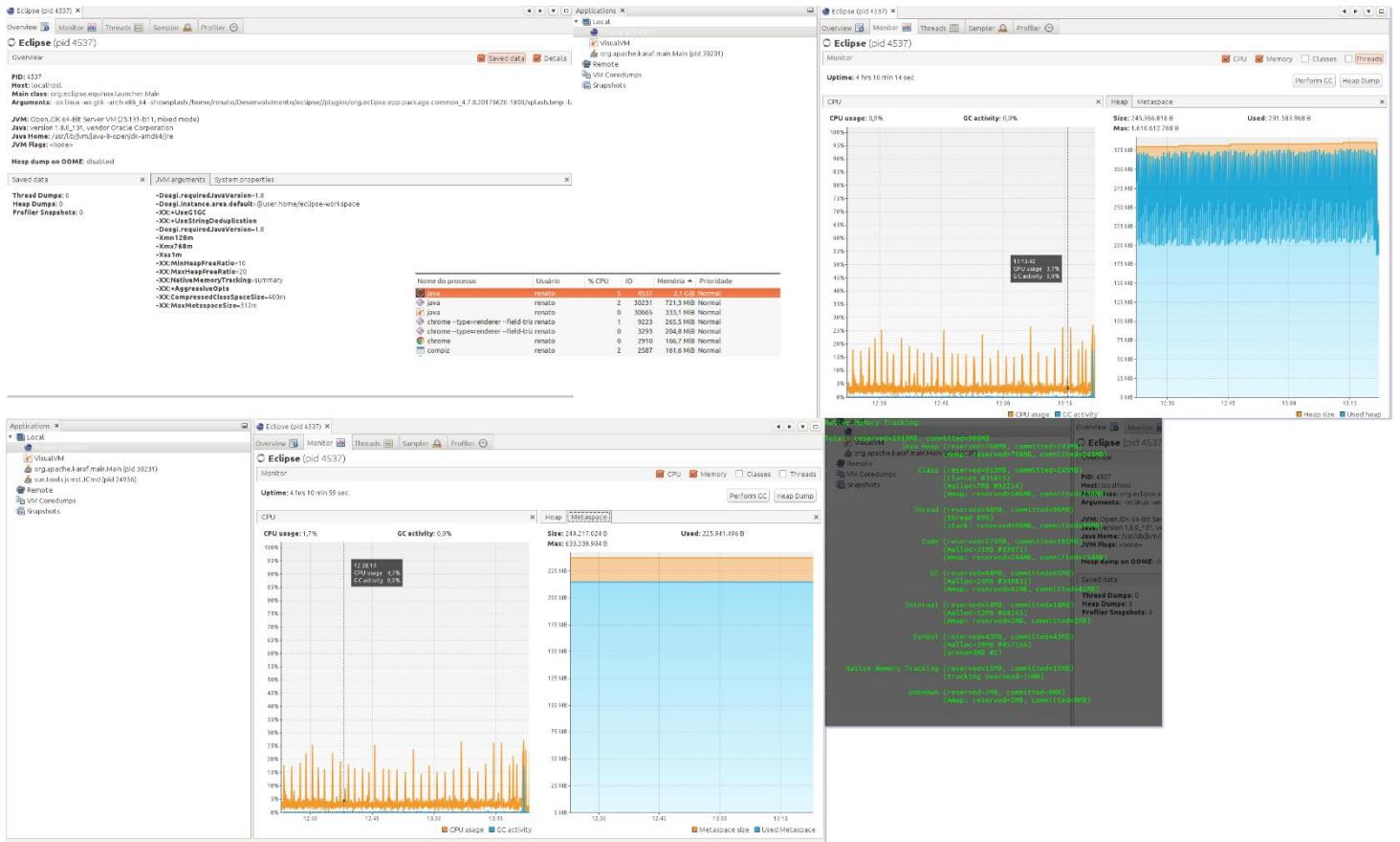


Fig 5 Consumo progresivo de memoria en Eclipse debido a la gestión ineficiente de la JVM



ECLIPSE: Presenta una interfaz inicial más espartana que NetBeans prácticamente hay que armar el entorno instalando plugins para cubrir necesidades básicas, lo cual incrementa la complejidad de uso. Esto resulta en una curva de aprendizaje más pronunciada, especialmente para novatos, Eclipse puede ser percibido como pesado y poco “moderno” su interfaz gráfica se considera anticuada en comparación con IDEs contemporáneos, Eclipse sacrifica facilidad de uso a cambio de poder de personalización, lo que puede dificultar su adopción inicial.

The screenshot shows the Eclipse IDE interface with the following components visible:

- Top Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Quick Shell, Window, Help.
- Left Sidebar:** Package Explorer, Outline, AI Chat, AI Chat X, AI Chat X (active), Discuss, AI Chat X (inactive).
- Middle Area:** A code editor window titled "Activator.java X" containing Java code for the "Activator" class. The code includes methods like `start`, `getDefault()`, and `getPromptsPreferencePresenter()`. A tooltip from the AI Chat perspective highlights a typo in the method name.
- Bottom Area:** Boot Dashboard, local.

```

1.22 package com.github.gradusnikov.eclipse.plugin.assista.main;
2.21
3.20 import java.util.Objects;
4.19
5.18 public class Activator extends AbstractUIPlugin
6.17 {
7.16     private static Activator plugin = null;
8.15
9.14 @Override
10.13     public void start(BundleContext context) throws Exception
11.12     {
12.11         super.start(context);
13.10         plugin = this;
14.9
15.8
16.7
17.6
18.5
19.4
20.3
21.2
22.1
23.14
24.13     public void start(BundleContext context) throws Exception
25.12     {
26.11         super.start(context);
27.10         plugin = this;
28.9
29.8
30.7
31.6
32.5
33.4
34.3
35.2
36.1
37.1
38.1
39.1
40.1
41.1
42.1
43.1
44.1
45.1
46.1
47.1
48.1
49.1
50.1
51.14
52.15
53.16
54.17
55.18
56.19
57.20
58.21
59.22
60.23
61.24
62.25
63.26
64.27
1.22 package com.github.gradusnikov.eclipse.plugin.assista.main;
2.21
3.20 import java.util.Objects;
4.19
5.18 public class Activator extends AbstractUIPlugin
6.17 {
7.16     private static Activator plugin = null;
8.15
9.14 @Override
10.13     public void start(BundleContext context) throws Exception
11.12     {
12.11         super.start(context);
13.10         plugin = this;
14.9
15.8
16.7
17.6
18.5
19.4
20.3
21.2
22.1
23.14
24.13     public void start(BundleContext context) throws Exception
25.12     {
26.11         super.start(context);
27.10         plugin = this;
28.9
29.8
30.7
31.6
32.5
33.4
34.3
35.2
36.1
37.1
38.1
39.1
40.1
41.1
42.1
43.1
44.1
45.1
46.1
47.1
48.1
49.1
50.1
51.14
52.15
53.16
54.17
55.18
56.19
57.20
58.21
59.22
60.23
61.24
62.25
63.26
64.27
    }
}
  
```

Fig 6 Pantalla que incluye plugin marketplace / extensiones instaladas, zona de código y menús múltiples.

AMENAZAS

COMPETENCIA DE OTROS IDES: El ecosistema de desarrollo Java y multilenguaje es muy competitivo. Encuestas recientes muestran que IntelliJ IDEA domina ampliamente el mercado de IDEs Java, seguido por Eclipse, mientras que Visual Studio Code ha superado el uso de NetBeans.

OBSOLESCENCIA TECNOLÓGICA: El ritmo de cambio en frameworks y lenguajes es veloz. Si Eclipse o NetBeans retrasan su adopción de nuevas versiones de Java u otros lenguajes, pueden quedar rezagados. Por ejemplo, aunque Eclipse continuamente lanza versiones trimestrales con soporte a Java 24, su interfaz gráfica mantiene un diseño clásico.

CAMBIOS EN EL MERCADO Y TECNOLOGÍAS: Evoluciones externas también representan amenazas. La creciente migración a entornos ligeros basados en navegador IDE en la nube y la integración de inteligencia artificial podrían desplazar parcialmente a los IDEs tradicionales. Como proyectos de código abierto, dependen de la comunidad, una caída en contribuciones. también, la aparición de nuevas plataformas de desarrollo o la discontinuación de tecnologías afectarían a ambos IDEs. En conjunto, Eclipse y NetBeans deben lidiar con un entorno tecnológico en rápido cambio y con alternativas comerciales muy competitivas.



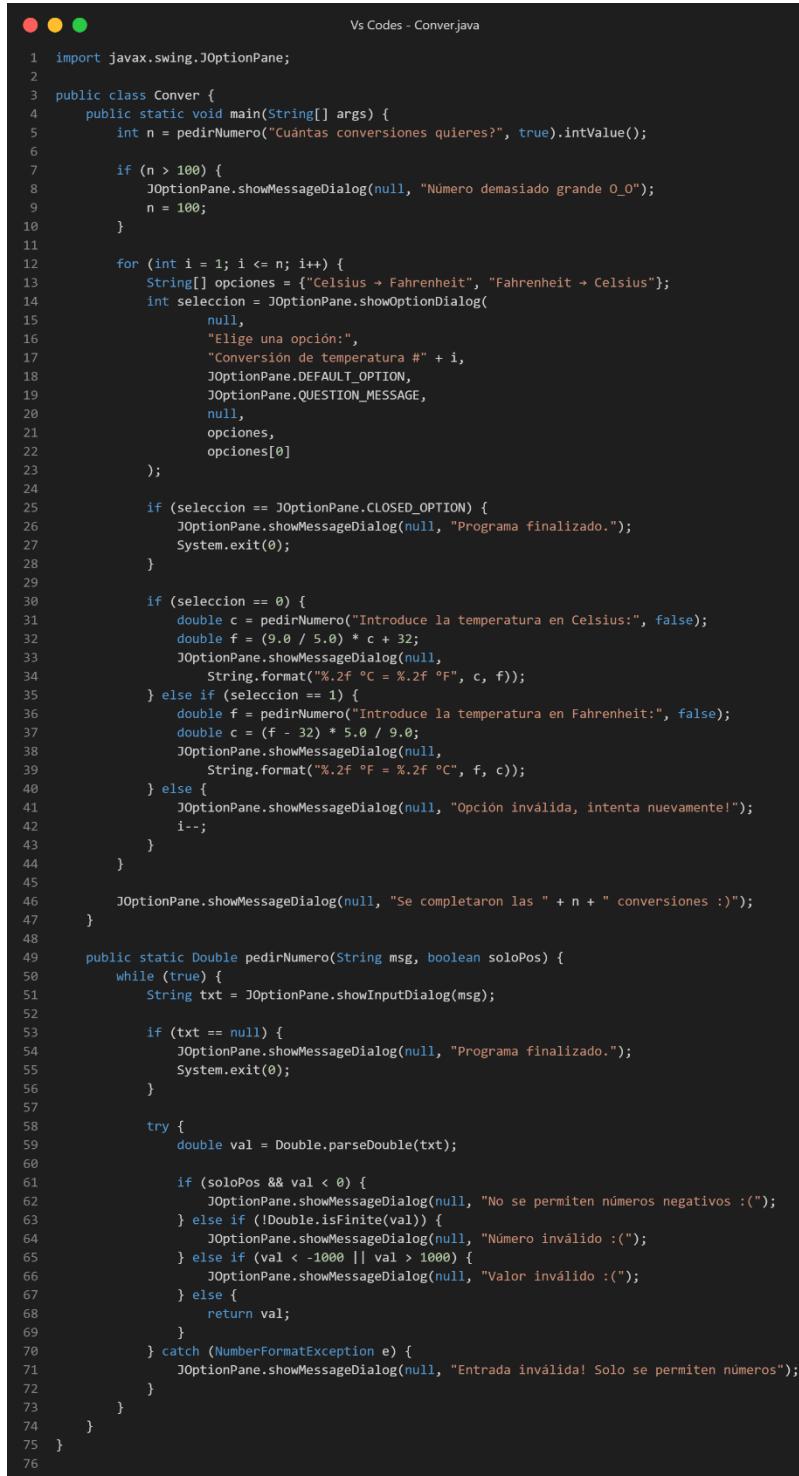
Fig 7 Competencia directa entre IDEs populares en el mercado de Java.

PARTE II

DESARROLLO

Problema: Escribir un método que convierta una temperatura dada en grados Celsius a grados Fahrenheit (viceversa); la fórmula de conversión es: $F = \frac{9}{5} * C + 32$.

Utilice JOptionPane para el desarrollo de su código, elija su IDE correspondiente. Controle la cantidad de veces que se va a ejecutar con un ciclo FOR.



```
1 import javax.swing.JOptionPane;
2
3 public class Conver {
4     public static void main(String[] args) {
5         int n = pedirNumero("Cuántas conversiones quieres?", true).intValue();
6
7         if (n > 100) {
8             JOptionPane.showMessageDialog(null, "Número demasiado grande 0_0");
9             n = 100;
10        }
11
12        for (int i = 1; i <= n; i++) {
13            String[] opciones = {"Celsius → Fahrenheit", "Fahrenheit → Celsius"};
14            int seleccion = JOptionPane.showOptionDialog(
15                null,
16                "Elige una opción:",
17                "Conversion de temperatura # " + i,
18                JOptionPane.DEFAULT_OPTION,
19                JOptionPane.QUESTION_MESSAGE,
20                null,
21                opciones,
22                opciones[0]
23            );
24
25            if (seleccion == JOptionPane.CLOSED_OPTION) {
26                JOptionPane.showMessageDialog(null, "Programa finalizado.");
27                System.exit(0);
28            }
29
30            if (seleccion == 0) {
31                double c = pedirNumero("Introduce la temperatura en Celsius:", false);
32                double f = (9.0 / 5.0) * c + 32;
33                JOptionPane.showMessageDialog(null,
34                    String.format("%2f ° = %2f °F", c, f));
35            } else if (seleccion == 1) {
36                double f = pedirNumero("Introduce la temperatura en Fahrenheit:", false);
37                double c = (f - 32) * 5.0 / 9.0;
38                JOptionPane.showMessageDialog(null,
39                    String.format("%2f °F = %2f °C", f, c));
40            } else {
41                JOptionPane.showMessageDialog(null, "Opción inválida, intenta nuevamente!");
42                i--;
43            }
44        }
45
46        JOptionPane.showMessageDialog(null, "Se completaron las " + n + " conversiones :)");
47    }
48
49    public static Double pedirNumero(String msg, boolean soloPos) {
50        while (true) {
51            String txt = JOptionPane.showInputDialog(msg);
52
53            if (txt == null) {
54                JOptionPane.showMessageDialog(null, "Programa finalizado.");
55                System.exit(0);
56            }
57
58            try {
59                double val = Double.parseDouble(txt);
60
61                if (soloPos && val < 0) {
62                    JOptionPane.showMessageDialog(null, "No se permiten números negativos :(");
63                } else if (!Double.isFinite(val)) {
64                    JOptionPane.showMessageDialog(null, "Número inválido :(");
65                } else if (val < -1000 || val > 1000) {
66                    JOptionPane.showMessageDialog(null, "Valor inválido :(");
67                } else {
68                    return val;
69                }
70            } catch (NumberFormatException e) {
71                JOptionPane.showMessageDialog(null, "Entrada inválida! Solo se permiten números");
72            }
73        }
74    }
75}
```

PARTE III

CASO DE ESTUDIO

PROCEDIMIENTO:

Teniendo en cuenta el siguiente plano de oficina (Figura 1), confeccione el esquema de RED LAN. Este esquema es la parte administrativa del Colegio AB. Todos los espacios deben tener conectividad (inalámbrica / cableado).

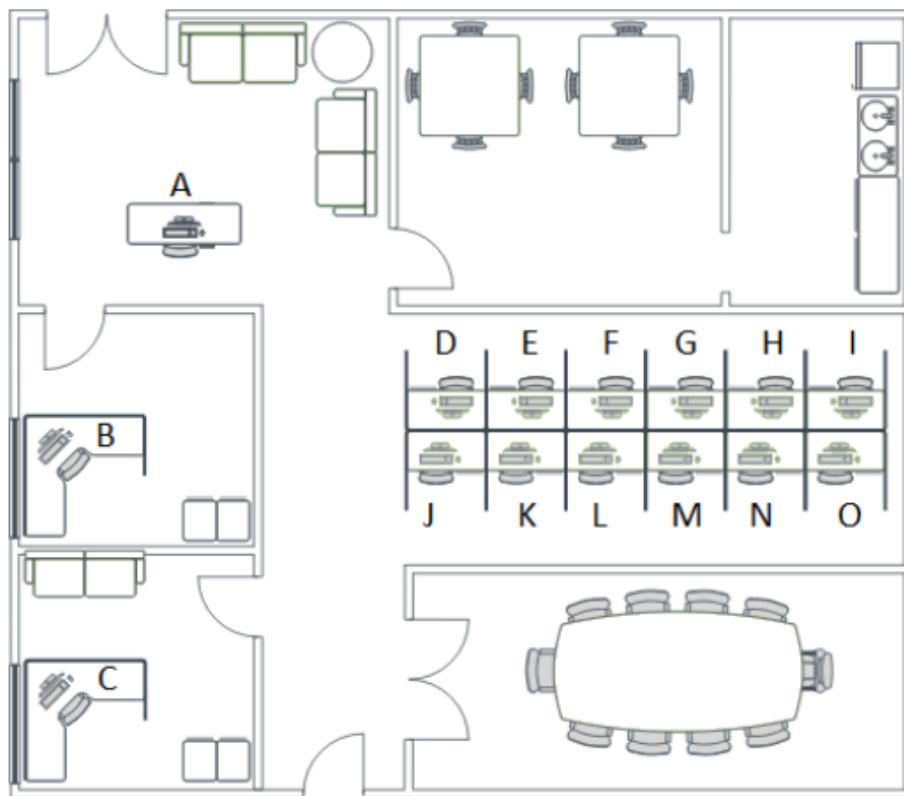


Fig 8 Plano de Oficina del Colegio AB

Requerimientos:

COLEGIO	EQUIPOS	IP	OBSERVACIÓN
AB	2 IMPRESORA, 1 SERVIDOR, 6 LAPTOP, 3 PC, 1 SWITCH 2960 DE 24 PUERTOS, 1 ROUTER300N, 2 EXTENSOR DE SEÑAL.	IP=10.10.10.10; MR=255.255.255.0, PE=10.10.10.1, DNS1=8.8.8.8, DNS2=8.8.4.4	PROVEEDOR TLK: ISP 1

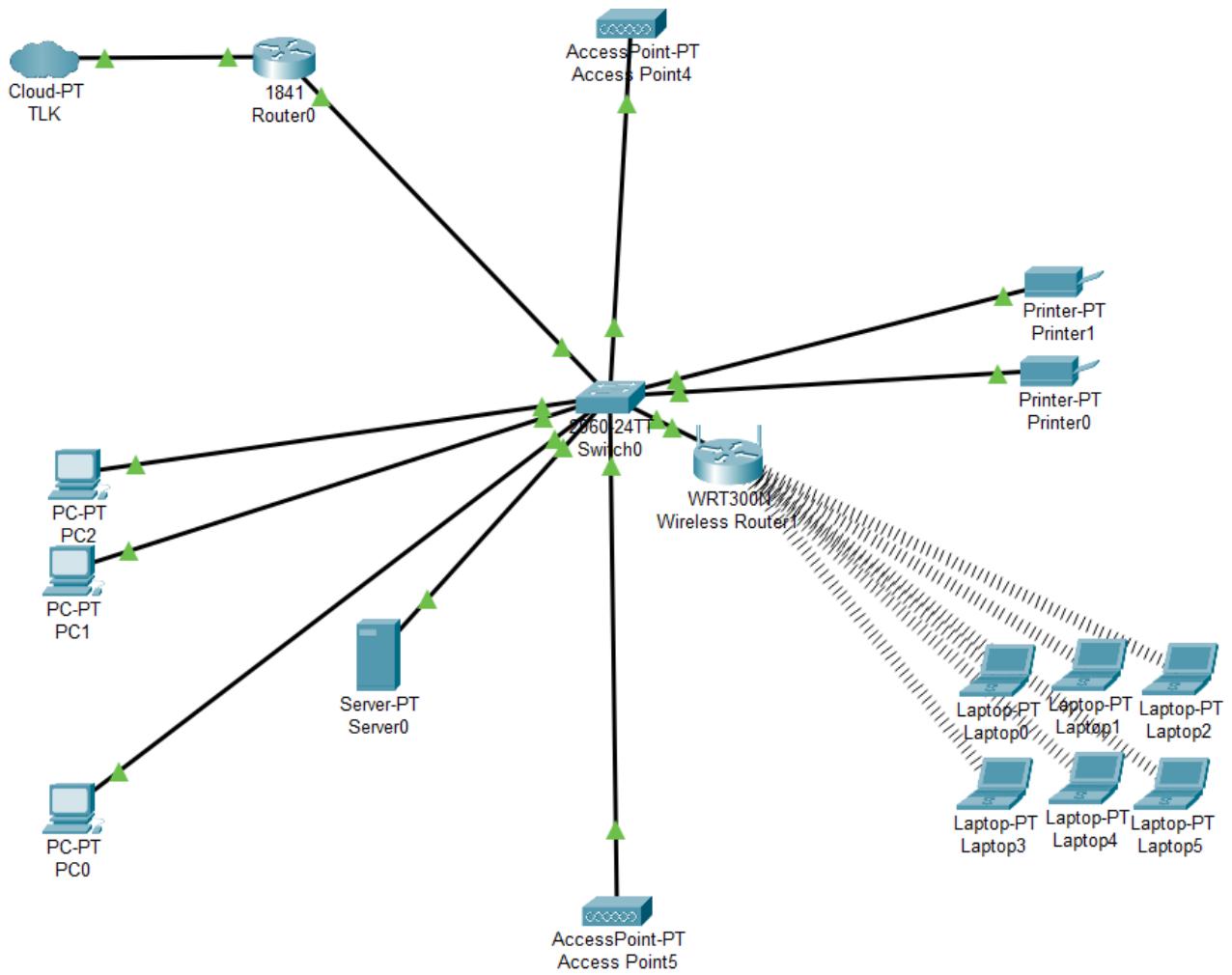


Fig 11 Plano de Oficina del Colegio AB simulado Logical

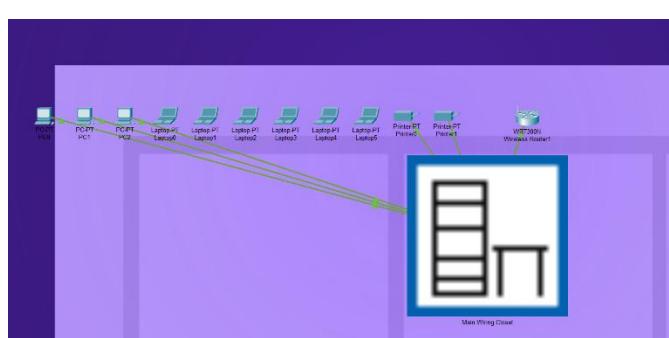


Fig 10 Plano de Oficina del Colegio AB simulado Physical 1



Fig 9 Plano de Oficina del Colegio AB simulado Physical 2

BIBLIOGRAFÍA

- Baeldung. (2024, 3 5). *The state of Java in 2024: IDEs*. Retrieved from Baeldung:
<https://www.baeldung.com/java-ides-in-2024>
- Foundation, E. (2025). *Eclipse IDE for Java Developers*. Retrieved from Eclipse Foundation:
<https://www.eclipse.org/ide/>
- JetBrains. (2024). *State of Developer Ecosystem 2024*. Retrieved from JetBrains:
<https://www.jetbrains.com/lp/devecosystem-2024/>
- NetBeans, A. (2025). *NetBeans IDE documentation*. Retrieved from Apache Software Foundation:
<https://netbeans.apache.org/>
- Overflow, S. (2024). *Developer Survey Results*. Retrieved from Stack Overflow:
<https://survey.stackoverflow.co/2024>
- Software, P. (2023). *The Rise and Fall of NetBeans: An IDE Market Analysis*. Retrieved from Perforce Reports: <https://www.perforce.com/blog>
- VMware. (2024). *Spring Tools Suite for Eclipse*. Retrieved from VMware, Inc.: <https://spring.io/tools>