Si el término “programación extrema” te hace pensar en los *X games* y los deportes extremos, no estás errado. De hecho, se trata de un proceso de programación bastante riguroso. Al igual que otros métodos ágiles de desarrollo de software, la programación extrema utiliza un enfoque adaptable y basado en pruebas para la ingeniería de software. Sin embargo, a diferencia de otros métodos, tiene reglas y valores estrictos que rigen cómo se realiza el trabajo.

¿Qué es la programación extrema (XP)?

La programación extrema es una metodología ágil de gestión de proyectos que se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. La estructura del proceso está determinada por 5 valores fundamentales, 5 reglas y 12 prácticas de XP (que detallaremos más adelante en este artículo).

Al igual que otras [metodologías ágiles](https://asana.com/es/resources/agile-methodology), la programación extrema es una método de desarrollo de software dividido en [sprints](https://asana.com/es/resources/sprint-planning-meeting) de trabajo. Los marcos ágiles siguen un [proceso iterativo](https://asana.com/resources/iterative-process), en el que se completa y revisa el marco al final de cada sprint, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la [eficiencia](https://asana.com/es/resources/ways-improve-team-efficiency-work') máxima. Al igual que otros métodos ágiles, el diseño de la programación extrema permite a los desarrolladores responder a las solicitudes de los clientes, adaptarse y realizar cambios en tiempo real. Sin embargo, la programación extrema es mucho más disciplinada; realiza revisiones de código frecuentes y pruebas unitarias para realizar cambios rápidamente. Además es muy creativa y [colaborativa](https://asana.com/es/resources/collaboration-in-the-workplace), ya que promueve el trabajo en equipo durante todas las etapas de desarrollo.

Programación extrema vs. Scrum

[Scrum](https://asana.com/es/resources/what-is-scrum) es otro tipo común de metodología ágil gestionada por un [Scrum master](https://asana.com/es/resources/scrum-master). Al igual que la programación extrema, organiza sprints basados ​​en historias de usuarios para desarrollar funciones nuevas de productos o de software. Sin embargo, el método XP es mucho más rígido que el método Scrum, ya que tiene reglas y pautas estrictas que promueven intercambios constantes entre desarrolladores y clientes. Además, puedes aplicar Scrum para cualquier proceso que requiera iteración y aportes del cliente, mientras que solo usarías la programación extrema para la programación.

¿Quién desarrolló la programación extrema?

Los [orígenes de XP](https://www.historicprojects.com/Extreme_Programming_XP.html#:~:text=Kent%20Beck%20(1961%2D)%20developed,company%20acquired%20Daimler%2DBenz)) se remontan a fines de la década de 1990, cuando Kent Beck la creó para gestionar el desarrollo de un sistema de software de nómina para Chrysler llamado Proyecto C3. El objetivo al implementar la programación extrema era (y sigue siendo) eliminar la resistencia a cambiar el código en un proyecto de desarrollo. En los métodos de desarrollo de software más tradicionales, es muy común que el código no se cambie una vez que está escrito (excepto para la depuración). Con la programación extrema, en cambio, el código se examina con tanto detalle que los desarrolladores pueden decidir modificarlo por completo luego de una sola iteración.

¿Cuándo deberías implementar la programación extrema?

Como la programación extrema se centra en el desarrollo de software, suele ser implementada solamente por los equipos de ingeniería. Incluso los equipos de software, suelen usarla únicamente para determinadas configuraciones. Para obtener el máximo beneficio de la programación extrema, recomendamos usarla en los siguientes casos:

* Para gestionar un equipo más pequeño. Debido a su naturaleza altamente colaborativa, la programación extrema funciona mejor en equipos pequeños de menos de diez personas.
* Si estás constantemente en contacto con tus clientes. La programación extrema incorpora los requisitos de los clientes a lo largo del proceso de desarrollo y también se basa en ellos para las pruebas y aprobaciones.
* Si trabajas con un equipo [flexible](https://asana.com/es/resources/workplace-adaptability-skills) que pueda aceptar el cambio (sin resentimientos). Dada su propia naturaleza, la programación extrema a menudo requerirá que todo el equipo deseche todo su arduo trabajo. Algunas reglas también permiten que algunos miembros del equipo realicen cambios en cualquier momento, lo que supondría un problema si los demás compañeros del equipo se lo toman como algo personal.
* Si dominas los aspectos técnicos de la codificación. La programación extrema no es para principiantes ya que necesitas poder trabajar e implementar cambios rápidamente.

Ciclo de vida de la programación extrema (XP)

El ciclo de vida de XP fomenta la integración continua, ya que requiere que los miembros del equipo trabajen casi constantemente, cada hora o todos los días. Sin embargo, el ciclo de vida completo se estructura de la siguiente manera:

* Extraer trabajos sin finalizar de las historias de usuarios
* Priorizar los elementos más importantes
* Comenzar con la planificación iterativa
* Incorporar un plan realista
* Mantener una comunicación constante con todas las partes interesadas y empoderar al equipo
* Presentar el trabajo
* Recibir comentarios
* Regresar a la etapa de planificación iterativa y repetir si es necesario

Los 5 valores de la programación extrema (XP)

La programación extrema está impulsada por el valor. En lugar de usar motivadores externos, la programación extrema permite que tu equipo trabaje de una manera más sencilla (priorizando la simplicidad y la colaboración sobre diseños complejos), basándose siempre en estos cinco valores.

1. Simplicidad

Antes de empezar cualquier trabajo de programación extrema, debes hacerte la siguiente pregunta: ¿Cuál es el proceso más simple y que también funciona? El concepto “que también funciona” es un diferenciador clave ya que lo más simple no siempre es práctico o efectivo. En la programación extrema, tu atención se centra en realizar primero el trabajo más importante. Esto significa que debes buscar un proyecto simple que sabes que puedes lograr.

2. Comunicación

XP se basa en una respuesta rápida y una [comunicación efectiva](https://asana.com/es/resources/effective-communication-workplace). Para trabajar de manera efectiva, el equipo debe ser abierto y ‌honesto entre sí. Cuando surgen problemas, se espera que todos aporten sus comentarios e ideas, ya que probablemente alguno de ellos ya tenga una solución adecuada. Y si no la tiene, podrán resolver el problema más rápidamente como grupo de lo que podrían hacerlo solos.

[Lee: Cómo mejorar las comunicaciones del equipo: 6 estrategias y muchos consejos útiles](https://asana.com/es/resources/team-communication)

3. Comentarios

Al igual que otras metodologías ágiles, el método XP incorpora comentarios e [historias de usuarios](https://asana.com/es/resources/user-stories) directamente en el proceso. El enfoque de XP es producir trabajo de forma rápida y sencilla, para luego compartir los resultados para obtener comentarios de forma casi inmediata. Por eso, los desarrolladores están en contacto casi constante con los clientes durante todo el proceso. En la programación extrema, se lanzan varias versiones con regularidad para obtener información nueva desde el primer momento y con frecuencia. Podrás adaptar el proceso (en lugar del proyecto) para incorporar los comentarios recibidos de clientes. Por ejemplo, si los comentarios ayudan a reducir el tiempo de retraso innecesario, podrás ajustar el proceso para que los desarrolladores trabajen para mejorar el tiempo de retraso en lugar de ajustar el proyecto completo.

[Lee: ¿Te resulta difícil brindar comentarios? Estos 20 consejos son ideales para ti](https://asana.com/es/resources/tips-giving-feedback)

4. Valentía

Para implementar la programación extrema, se requiere de mucha valentía. Siempre se espera que seas honesto al brindar actualizaciones al equipo sobre tu progreso, lo que puede dejarte en una posición de vulnerabilidad. Si no cumples con una fecha de entrega en la programación extrema, es probable que al líder de tu equipo no le interese analizar los motivos. En cambio, le dirías que no cumpliste con la fecha de entrega, te responsabilizarías por ello y te pondrías a trabajar nuevamente.

Si eres líder de un equipo, tu responsabilidad al comienzo del proceso de XP será establecer las expectativas de éxito y definir lo que será “el trabajo terminado”. A menudo suele haber poca planificación para lidiar con el fracaso ya que el equipo se centra principalmente en el éxito. Sin embargo, puede ser un poco alarmante ya que sabemos que las cosas no siempre suceden como se planifican. Pero si surgen cambios durante el proceso de XP, se espera que el equipo se adapte y acompañe estos cambios.

5. Respeto

Teniendo en cuenta que las comunicaciones y la honestidad son prioridad en el método XP, tiene sentido que el respeto sea una virtud esencial. Para que los equipos se comuniquen y colaboren de manera efectiva, deben aprender a estar en desacuerdo. Pero hay maneras de hacerlo amablemente. El respeto es una base importante que promueve la bondad y la confianza, incluso cuando se expresan las opiniones con total honestidad. Para la programación extrema, estas son las expectativas:

* Respeto mutuo entre los clientes y el equipo de desarrollo.
* Respeto mutuo entre los miembros del equipo.
* El reconocimiento de que todos en el equipo aportan algo valioso al proyecto.

5 reglas de la metodología de programación extrema (XP)

Los valores de la programación extrema son los aspectos más filosóficos. Las reglas, por otro lado, son los usos prácticos de cómo se realiza el trabajo. Necesitarás contar con ambos aspectos para gestionar un equipo XP efectivo.

1. Planificación

Durante las [etapas de planificación](https://asana.com/es/resources/sprint-planning-meeting) de la programación extrema, deberás determinar si el proyecto es viable y si se adapta al método XP. Para hacer esto, deberás evaluar:

* Las historias de usuarios para confirmar que coinciden con el valor de simplicidad y garantizar que el cliente esté disponible para participar del proceso. Si la historia del usuario es más compleja o fue creada por un cliente anónimo, es probable que no sea adecuada para el método XP.
* El valor comercial y el nivel de prioridad del proyecto para asegurarte de que esté alineado con la idea de “realizar el trabajo más importante primero”.
* La etapa de desarrollo en la que te encuentras. La programación extrema se adapta mejor al desarrollo en sus etapas iniciales y no será un método tan efectivo para iteraciones posteriores.

Una vez que hayas confirmado que el proyecto es viable para implementar la programación extrema, ‌te recomendamos crear un cronograma de lanzamiento. Sin embargo, es importante tener en cuenta que debes lanzarlo tan pronto sea posible y con frecuencia para poder obtener comentarios. Para hacer esto:

* Divide el proyecto en iteraciones y crea un plan para cada una.
* Establece plazos realistas y un ritmo de trabajo sostenible.
* Comparte actualizaciones a medida que ocurren para permitir que el equipo sea lo más honesto y transparente posible.
* Comparte actualizaciones en tiempo real para ayudar al equipo a ‌identificar, adaptar y realizar cambios más rápidamente.
* Usa una [herramienta de gestión de proyectos](https://asana.com/es/product) para crear un [tablero Kanban](https://asana.com/es/resources/what-is-project-portfolio-management) o un cronograma para poder dar seguimiento al progreso en tiempo real.

2. Gestión

Uno de los elementos clave de la programación extrema es el espacio de trabajo. Los puristas de XP recomiendan usar un espacio de trabajo abierto donde todos los miembros del equipo puedan trabajar juntos. Debido a que la programación extrema es altamente colaborativa, es una ventaja contar con un espacio donde puedan reunirse todos los colaboradores físicamente. Sin embargo, en estas épocas, esta no siempre es una solución práctica o factible. Si trabajas en un equipo remoto, considera usar una [plataforma](https://asana.com/es/product) que fomente el [trabajo asincrónico](https://asana.com/es/resources/synchronous-vs-asynchronous-communication) para la [colaboración remota](https://asana.com/es/resources/remote-collaboration). De esta manera, todos los miembros podrán trabajar juntos en el proyecto, incluso si no se encuentran en el mismo espacio físico.

Al igual que con otros métodos ágiles, puedes organizar [reuniones diarias de actualización](https://asana.com/es/templates/for/marketing/agile-daily-standup) para verificar el estado del trabajo y fomentar las comunicaciones abiertas y constantes. Te recomendamos implementar un ciclo semanal y un ciclo trimestral. Durante el ciclo trimestral, tu equipo y tú revisarán las historias que guiarán el trabajo a realizar. También analizarán los procesos de XP, para identificar brechas u oportunidades para realizar cambios. Luego, trabajarás en ciclos semanales, que comenzarán con una reunión con el cliente. El cliente elegirá la historia de usuario para que los programadores trabajen esa semana.

Como gerente o líder de equipo, tu atención deberá centrarse en mantener el progreso constante del trabajo, supervisar el ritmo, gestionar la carga de trabajo del equipo para que puedan trabajar en los errores o problemas a medida que surjan, o cambiar el proceso de XP para que se ajuste a tu proyecto e iteración actuales. Recuerda que el objetivo de la programación extrema es ser flexible y ágil, por lo que tu trabajo estará muy centrado en el trabajo actual del equipo y en poder responder rápidamente a cualquier cambio.

3. Diseño

Si recién incursionas en la programación extrema, te recomemos empezar con el diseño más simple posible, ya que las iteraciones posteriores incrementarán su complejidad. Intenta no agregar ninguna funcionalidad temprana durante esta etapa para mantener el proceso lo más sencillo posible.

Los equipos de metodología XP a menudo usan tarjetas de clase, responsabilidad y colaboración (CRC, por sus siglas en inglés) para mostrar cómo interactúa cada elemento en el diseño global. Al completar cada campo de la tarjeta, podrás obtener una imagen visual de cómo se relacionan e interactúan todas las funciones. Las tarjetas CRC incluyen:

* Clase (conjunto de elementos similares)
* Responsabilidades (según la clase)
* Colaboradores (clase que interactúa con el colaborador en cuestión)

Las tarjetas CRC son útiles para simular el proceso y detectar problemas potenciales. Independientemente del tipo de diseño, te recomendamos usar un sistema que permita reducir los cuellos de botella potenciales. Para esto, asegúrate de prevenir los riesgos de manera proactiva. Tan pronto como descubras una amenaza potencial, asigna uno o dos miembros del equipo para encontrar una solución en caso de que suceda.

[Lee: El proceso de gestión de riesgos de proyectos en 6 pasos claros](https://asana.com/es/resources/project-risk-management-process)

4. Codificación

Uno de los aspectos clave de la programación extrema es el contacto permanente que mantendrás con el cliente durante todo el proceso de codificación. Esta interacción ‌te permite probar e incorporar comentarios dentro de cada iteración, en lugar de esperar hasta el final de cada sprint. Sin embargo, las reglas de codificación son bastante estrictas en el método XP. Algunas incluyen:

* Todo el código debe cumplir con el estándar de programación.
* Se deben realizar pruebas unitarias para definir los requisitos y desarrollar todos los aspectos del proyecto.
* La programación se realiza en parejas: dos desarrolladores trabajan juntos y simultáneamente en la misma computadora. El tiempo de programación sigue siendo el mismo; sin embargo, se duplica el enfoque para lograr resultados de la más alta calidad posible.
* Se deben usar integraciones continuas para agregar código nuevo y probarlo de inmediato.
* Solo un par de desarrolladores puede actualizar el código en un momento dado para reducir los errores.
* Propiedad colectiva del código: cualquier miembro del equipo puede cambiar el código en cualquier momento.

5. Prueba

Deberás realizar pruebas durante todo el proceso de programación extrema. Todo el código deberá someterse a pruebas unitarias antes de su lanzamiento. Si detectas errores durante estas pruebas, necesitarás crear pruebas adicionales para corregirlos. Más adelante‌, incorporarás la misma [historia de usuario](https://asana.com/es/resources/user-stories) en la que has estado trabajando en una prueba de aceptación. Durante esta prueba, el cliente examinará los resultados para comprobar que has implementado correctamente la historia de usuario al producto.

Raeburn, A. (2025, 13 febrero). ¿Qué es la programación extrema (XP)? [2025] • Asana. *Asana*. https://asana.com/es/resources/extreme-programming-xp

**¿Qué es MySQL?**

MySQL es el sistema de gestión de bases de datos de código abierto más popular del mundo. Las [bases de datos](https://www.oracle.com/mx/database/what-is-database/) son los repositorios de información esencial para todas las aplicaciones de software. Por ejemplo, cada vez que alguien realiza una búsqueda en Internet, inicia sesión en una cuenta o completa una transacción, una base de datos almacena la información para poder acceder a ella en el futuro. MySQL sobresale en esta tarea.

SQL, que significa [lenguaje de consulta estructurado (Structured Query Language)](https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/Introduction-to-Oracle-SQL.html#GUID-049B7AE8-11E1-4110-B3E4-D117907D77AC), es un lenguaje de programación que se utiliza para recuperar, actualizar, suprimir y manipular datos en bases de datos relacionales. MySQL se pronuncia oficialmente "My es-kiu-el", pero "my sequel" es una variación común. Como sugiere el nombre, MySQL es una base de datos relacional basada en [SQL](https://www.oracle.com/mx/database/technologies/appdev/sql.html) diseñada para almacenar y gestionar datos estructurados. En los últimos años, sin embargo, Oracle añadió soporte adicional, incluyendo el popular tipo de datos JSON.

**Conclusiones clave**

* En 2024, MySQL mantiene su posición como la base de datos de código abierto más popular del mundo.
* Como sistema de base de datos relacional, MySQL almacena datos en filas y columnas definidas por esquemas.
* MySQL deriva parte de su nombre del lenguaje SQL, que se usa para gestionar y consultar datos en bases de datos.
* MySQL ofrece transacciones ACID completas y puede manejar un alto volumen de conexiones simultáneas.

**MySQL en detalle**

MySQL es un RDBMS de código abierto que utiliza SQL para crear y gestionar bases de datos. Como base de datos relacional, MySQL almacena datos en tablas de filas y columnas organizadas en esquemas. Un esquema define cómo se organizan y almacenan los datos y describe la relación entre varias tablas. Con este formato, los desarrolladores pueden almacenar, recuperar y analizar fácilmente muchos tipos de datos, incluidos texto simple, números, fechas, horas y, más recientemente, JSON y [vectores](https://blogs.oracle.com/mysql/post/become-an-ai-developer-with-mysql-heatwave).

Dado que MySQL es de código abierto, incluye numerosas características desarrolladas en estrecha cooperación con una comunidad de usuarios durante casi 30 años. Dos capacidades en las que los desarrolladores confían son el soporte de transacciones ACID y la capacidad de escalado de MySQL. ACID significa "atomicidad, consistencia, aislamiento y durabilidad" (o atomicity, consistency, isolation y durability), las cuatro propiedades que garantizan que las transacciones en bases de datos se procesen de manera confiable y precisa. Con las transacciones ACID, MySQL puede garantizar que todas las modificaciones de datos se realicen de manera coherente y fiable, incluso en caso de un fallo del sistema. MySQL puede escalarse para admitir bases de datos muy grandes y puede manejar un alto volumen de conexiones simultáneas.

El rendimiento de MySQL, su facilidad de uso y bajo costo, combinado con su capacidad de escalar de manera confiable a medida que un negocio crece, lo han convertido en la base de datos de código abierto más popular del mundo.

**MySQL: sus diferencias con SQL**

El acrónimo "SQL" significa "lenguaje de consulta estructurado", un tipo de lenguaje de programación que se utiliza para manipular datos en una base de datos. MySQL utiliza el lenguaje SQL para gestionar y consultar datos en bases de datos y, por lo tanto, utiliza el acrónimo como parte de su nombre. Si tienes datos almacenados en un RDBMS MySQL, puedes escribir instrucciones SQL simples para añadir, buscar, analizar y recuperar datos.

**Entendiendo MySQL: características y popularidad**

La capacidad de MySQL para almacenar y analizar eficientemente grandes cantidades de datos significa que puede ayudar con tareas tan variadas como informar decisiones comerciales complejas o encontrar un restaurante local para una cita. A continuación, un vistazo a las principales funcionalidades que hacen que MySQL sea tan omnipresente en el panorama tecnológico actual.

**Un sistema completo de base de datos relacional**

MySQL es conocido como un sistema de gestión de bases de datos flexible y fácil de usar. Lo usan desde desarrolladores independientes que toman una base de datos de código abierto para un pequeño proyecto, hasta los sitios web y aplicaciones más visitados del mundo. MySQL ha estado evolucionando para satisfacer la demanda durante casi 30 años y ofrece transacciones ACID que garantizan que las modificaciones de datos se realicen de manera coherente, incluso cuando admite un alto volumen de conexiones simultáneas.

**La ventaja del código abierto de MySQL**

MySQL es [de código abierto](https://www.oracle.com/mx/developer/open-source-developers/), lo que significa que cualquiera puede descargar el software de MySQL de internet y usarlo sin costo. Las organizaciones también pueden cambiar su código fuente para adaptarlo a sus necesidades. El software de MySQL utiliza la [GNU General Public License](https://www.gnu.org/licenses/license-list.html) (GPL), que es un conjunto común de reglas para definir qué se puede o no hacer con el software en diversas situaciones. Si una organización se siente incómoda con la GNU GPL o desea incorporar código de MySQL en una aplicación comercial, puede comprar una versión con licencia comercial. Consulta la página de [MySQL Legal Policies](https://www.mysql.com/about/legal/) para obtener más información sobre licencias.

**Por qué los desarrolladores prefieren el rendimiento y la flexibilidad de MySQL**

MySQL es conocido por ser fácil de configurar y usar, pero lo suficientemente fiable y escalable para organizaciones con conjuntos de datos muy grandes y un gran número de usuarios. La arquitectura de replicación nativa de MySQL permite a organizaciones como Facebook escalar aplicaciones para admitir miles de millones de usuarios.

Otros factores clave de la popularidad de MySQL incluyen recursos de aprendizaje abundantes y la vibrante [comunidad global](https://dev.mysql.com/community/) del software.

**¿Cómo funciona MySQL?**

Cada aplicación de software necesita un repositorio para almacenar datos, de modo que la información pueda ser accesada, actualizada y analizada en el futuro. Una [base de datos relacional](https://www.oracle.com/mx/database/what-is-a-relational-database/) como MySQL almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. La estructura de la base de datos está organizada en archivos optimizados para poder acceder a ellos rápidamente. Este modelo de datos lógico, con objetos como tablas de datos, vistas, filas y columnas, ofrece a los desarrolladores y administradores de bases de datos un entorno de programación flexible. Ellos pueden establecer reglas que gobiernen las relaciones entre diferentes campos de datos, como uno a uno, uno a muchos, único, requerido u opcional, y añadir "punteros" entre diferentes tablas. El sistema aplica estas reglas para que, con una base de datos bien diseñada, una aplicación nunca vea datos que sean inconsistentes, duplicados, huérfanos o desactualizados.

MySQL Database es un sistema cliente/servidor que consta de un servidor SQL multihilo que admite diferentes backends, varios programas y bibliotecas cliente, una selección de herramientas administrativas y una amplia variedad de interfaces de programación de aplicaciones (API). MySQL está disponible como una biblioteca multihilo integrada que los desarrolladores pueden vincular a aplicaciones para obtener un producto autónomo más pequeño, rápido y fácil de gestionar.

SQL es el lenguaje de programación estandarizado más común utilizado para acceder a bases de datos. Dependiendo del entorno de programación, un desarrollador podría introducir SQL directamente, por ejemplo, para generar informes. También es posible incrustar instrucciones SQL en código escrito en otro lenguaje de programación o usar una API específica de un lenguaje que oculte la sintaxis SQL.

**¿Por qué MySQL es importante?**

MySQL es importante debido a su ubicuidad y al papel fundamental que desempeñan las bases de datos a medida que la cantidad de datos crece exponencialmente y alimenta la inteligencia artificial. MySQL sustenta una amplia variedad de sitios web y aplicaciones, ayudando a empresas de todo el mundo a organizar, analizar y proteger sus datos.

Otros factores también contribuyen a la larga popularidad de MySQL.

* **Código abierto con fuerte apoyo comunitario**

Durante las casi tres décadas de MySQL como el principal RDBMS de código abierto, se ha desarrollado una vibrante comunidad global a su alrededor. Esto es importante porque la comunidad ofrece una gran cantidad de experiencia y recursos, como tutoriales, consejos en foros y más. Al probar el software en múltiples escenarios de uso, la comunidad también ha ayudado a descubrir y solucionar errores, haciendo que MySQL sea altamente confiable.

El intercambio de conocimientos, la resolución de problemas y la innovación continua de la comunidad de código abierto mantienen a los usuarios de MySQL a la vanguardia de los avances tecnológicos.

* **Alto rendimiento y fiabilidad**

MySQL se adapta a muchos entornos diferentes, desde proyectos individuales de desarrolladores hasta aplicaciones críticas que requieren una sólida estabilidad. Este RDBMS de código abierto puede manejar grandes volúmenes de datos y conexiones concurrentes, proporcionando operaciones ininterrumpidas incluso en circunstancias exigentes. Esto se debe en parte a los mecanismos robustos de replicación y conmutación por error de MySQL, que ayudan a minimizar el riesgo de pérdida de datos.

* **Facilidad de uso y compatibilidad**

MySQL es frecuentemente elogiado por su facilidad de uso y su amplia [compatibilidad](https://www.mysql.com/support/supportedplatforms/database.html) con plataformas tecnológicas y lenguajes de programación, incluyendo Java, Python, PHP y JavaScript. MySQL también admite la replicación entre versiones, por lo que una aplicación que use MySQL 5.7 puede replicarse fácilmente a MySQL 8.0.

Además, MySQL ofrece flexibilidad para desarrollar aplicaciones de base de datos tradicionales basadas en SQL y aplicaciones NoSQL sin esquemas. Esto permite a los desarrolladores combinar datos relacionales y documentos JSON en la misma base de datos y aplicación.

* **Rentabilidad y escalabilidad**

Dado que MySQL es de código abierto, está disponible gratuitamente, excepto por el hardware en el que se ejecuta en las instalaciones y la capacitación necesaria para usarlo. Para lo último, una comunidad global de usuarios de MySQL ofrece acceso rentable a recursos de aprendizaje y experiencia en resolución de problemas. Oracle también ofrece [una amplia gama de cursos de capacitación](https://www.mysql.com/training/).

Cuando sea necesario escalar, MySQL admite multihilos para manejar grandes cantidades de datos de manera eficiente. Las características de conmutación por error automatizada ayudan a reducir los costos potenciales de tiempos de inactividad no planificados.

**Ventajas de MySQL**

MySQL es rápido, confiable, ampliable y fácil de utilizar. Originalmente fue desarrollado para manejar bases de datos grandes rápidamente y se ha utilizado en entornos de producción altamente exigentes durante muchos años. MySQL ofrece un conjunto rico y útil de funciones, y está en constante desarrollo por parte de Oracle, lo que le permite mantenerse al día con las demandas tecnológicas y empresariales. La conectividad, la velocidad y la seguridad de MySQL lo hacen muy adecuado para acceder a bases de datos en Internet.

Los beneficios clave de MySQL incluyen:

* **Facilidad de uso.** Los desarrolladores pueden instalar MySQL en minutos, y la base de datos es fácil de gestionar.
* **Confiabilidad.** MySQL es una de las bases de datos más maduras y ampliamente utilizadas. Ha sido probada en una amplia variedad de escenarios durante casi 30 años, incluyendo muchas de las empresas más grandes del mundo. Las organizaciones utilizan MySQL para ejecutar aplicaciones clave para el negocio debido a su confiabilidad.
* **Escalabilidad.** MySQL se escala para satisfacer las demandas de las aplicaciones más utilizadas. La arquitectura de replicación nativa de MySQL permite a organizaciones como Facebook, Netflix y Uber escalar aplicaciones para soportar decenas de millones de usuarios o más.
* **Rendimiento.** MySQL es un sistema de base de datos de alto rendimiento y sin administración probado, disponible en [una variedad de ediciones](https://www.mysql.com/products/) para satisfacer casi cualquier demanda. La solución basada en la nube [HeatWave MySQL](https://www.oracle.com/mx/mysql/) proporciona un rendimiento y relación precio-rendimiento incomparables, según los estándares de la industria como TPC-H, TPC-DS y CH-benCHmark.
* **Alto grado de disponibilidad.** MySQL ofrece un conjunto completo de tecnologías de replicación nativas e integradas para alta disponibilidad y recuperación ante desastres. Para aplicaciones críticas y compromisos de nivel de servicio, los clientes pueden lograr un objetivo de punto de recuperación cero (sin pérdida de datos) y un objetivo de tiempo de recuperación de cero segundos (conmutación automática por error).
* **Seguridad.** [La seguridad de datos](https://www.oracle.com/mx/security/database-security/what-is-data-security/) implica la protección y el cumplimiento de las normativas gubernamentales y del sector, como el Reglamento General de Protección de Datos de la Unión Europea, la Norma de Seguridad de Datos del Sector de Tarjetas de Pago, la Ley de Portabilidad y Responsabilidad de Seguros Médicos y las Guías de Implementación Técnica de Seguridad de la Agencia de Sistemas de Información de Defensa. [MySQL Enterprise Edition](https://www.oracle.com/mx/mysql/enterprise/) ofrece características avanzadas de seguridad, como autenticación/autorización, cifrado transparente de datos, auditoría, enmascaramiento de datos y un firewall de base de datos.
* **Flexibilidad.** MySQL Document Store ofrece a los usuarios la máxima flexibilidad en el desarrollo de aplicaciones de bases de datos tradicionales SQL y NoSQL sin esquemas. Los desarrolladores pueden combinar datos relacionales y documentos JSON en la misma base de datos y aplicación.

**Casos de uso de MySQL**

Los casos de uso de MySQL incluyen la gestión de datos de clientes y productos para sitios web de comercio electrónico, ayudar a los sistemas de gestión de contenido a servir contenido web, rastrear de manera segura transacciones y datos financieros, y potenciar sitios de redes sociales al almacenar perfiles de usuarios e interacciones.

La capacidad de MySQL para manejar grandes conjuntos de datos y consultas complejas lo convierte en una tecnología clave en diversas industrias y casos de uso, incluidos los siguientes:

* **Comercio electrónico.** Muchas de las aplicaciones de comercio electrónico más grandes del mundo, incluidas Uber y Booking.com, ejecutan sus sistemas transaccionales en MySQL. Es una elección popular para gestionar perfiles de usuarios, credenciales, contenido generado por usuarios y datos financieros, incluidos los pagos, además de detección de fraudes.
* **Redes sociales.** Facebook, X (anteriormente Twitter) y LinkedIn están entre las redes sociales más grandes del mundo, y todas dependen de MySQL.
* **Gestión de contenido.** A diferencia de las bases de datos de documentos de un solo propósito, MySQL permite tanto SQL como NoSQL en una sola base de datos. MySQL Document Store permite operaciones CRUD y la potencia de SQL para consultar datos de documentos JSON para informes y análisis.
* **SaaS e ISV.** Más de 2000 ISV, OEM y VAR, incluidos Ericsson e IBM, dependen de MySQL como la base de datos integrada para hacer sus aplicaciones, hardware y dispositivos más competitivos; llevar productos al mercado más rápido; y reducir sus costos de producción. MySQL también es la base de datos detrás de populares aplicaciones SaaS, como Zendesk. Otras aplicaciones populares que usan MySQL incluyen apps para juegos en línea, marketing digital, sistemas de punto de venta en retail y sistemas de monitoreo de internet de las cosas (IoT).
* **Aplicaciones locales con MySQL Enterprise Edition.** MySQL Enterprise Edition incluye el conjunto más completo de funciones avanzadas junto con herramientas de gestión y soporte técnico, permitiendo a las organizaciones alcanzar los niveles más altos de escalabilidad, seguridad, confiabilidad y tiempo de actividad de MySQL. Reduce el riesgo, el costo y la complejidad mediante el desarrollo, el despliegue y la gestión de aplicaciones MySQL críticas para los negocios. Ofrece funciones de seguridad, como MySQL Enterprise Backup, Monitor, Firewall, Audit, Transparent Data Encryption y Authentication, para ayudar a los clientes a proteger los datos y lograr el cumplimiento normativo y del sector.

MySQL es la base de datos de código abierto más popular del mundo y ocupa el segundo lugar entre todas las bases de datos, solo detrás de Oracle Database, por una razón. Es confiable y rápido, con un rendimiento del que dependen miles de sitios web y aplicaciones. Puede manejar grandes cantidades de datos y consultas sin ralentizarse. Cuenta con una comunidad de código abierto grande y entusiasta. Todo eso, sumado a su bajo costo y facilidad de uso, hacen de MySQL una elección principal para muchos desarrolladores y empresas.

Erickson, J. (2024, 29 agosto). *MySQL: Understanding What It Is and How It’s Used*. https://www.oracle.com/mx/mysql/what-is-mysql/

¿Qué es un diagrama entidad relación?

Un diagrama entidad relación (también conocido como diagrama ER o diagrama ERD o simplemente ERD) muestra cómo interactúan las entidades (personas, objetos y conceptos). Estos modelos conceptuales de datos ayudan a desarrolladores y diseñadores a visualizar las relaciones entre elementos clave del software.

reados por el informático Peter Chen en los años 70, los ERD son utilizados habitualmente por las empresas en el diseño de bases de datos. Sin embargo, los diagramas entidad relación también pueden representar las relaciones entre otros elementos de la empresa, como las funciones (por ejemplo, la relación de un product manager con un desarrollador), los objetos tangibles de la empresa (como un producto o servicio) y los objetos intangibles de la empresa (como el [product backlog](https://miro.com/es/plantillas/product-backlog/)). El diagrama en sí es un tipo de [diagrama de flujo](https://miro.com/es/diagrama-de-flujo/que-es-diagrama-de-flujo/). Es un modelo lógico que muestra cómo fluyen los datos de una entidad a otra. Con este formato fácil de seguir, los desarrolladores y diseñadores de software pueden visualizar claramente la estructura de un sistema.

Comprende claramente cómo están estructurados tus datos con un diagrama entidad relación. Con esta representación visual de tus datos, puedes identificar fácilmente las relaciones entre las diferentes entidades y tomar mejores decisiones sobre cómo gestionarlas. Obtén más información sobre cómo se utilizan estos diagramas o crea el tuyo haciendo un [diagrama entidad relación online](https://miro.com/es/diagrama/diagrama-entidad-relacion/).

Símbolos del diagrama entidad relación

Cada diagrama presenta una selección de símbolos de diagrama entidad relación. Estos son algunos de los símbolos más comunes:

Rectángulos

Los rectángulos representan las entidades padre. Suelen tener entidades débiles (rectángulos dobles) que dependen de ellas.

Óvalos

Los atributos se representan mediante óvalos. Describen las características de una entidad.

Rombos

Los rombos se utilizan para mostrar las relaciones entre entidades.

Flechas y líneas de conexión

Las líneas y flechas sirven para indicar las relaciones entre entidades.

Cardinalidades

Las cardinalidades indican la cantidad de información que fluye de una entidad a otra (más información sobre este tema más adelante).

¿Por qué es importante el diagrama ER en el diseño de bases de datos?

A la hora de lanzar un nuevo software, los desarrolladores y diseñadores de bases de datos necesitan una forma sencilla pero eficaz de visualizar la estructura. Necesitan saber cómo funcionan juntas las entidades, sus atributos y cómo se relacionan entre sí. Aquí es donde puede ayudar un modelo ER lógico. Un ERD permite a los desarrolladores mapear cada elemento clave de una nueva base de datos. Pueden ver cómo funcionará el software, probar ciertas relaciones de antemano y detectar cualquier laguna en el software antes de que se ponga en marcha. De este modo, el equipo de desarrollo tendrá un proceso más fluido en el momento de la puesta en marcha. La creación de un diagrama entidad relación también ayuda a los desarrolladores a detectar áreas de mejora en las bases de datos existentes. Facilita la visualización de las áreas en las que el software actual necesita mejoras. Los desarrolladores pueden entonces depurar las bases de datos para mejorar el software en el futuro.

¿Cuáles son los 3 componentes principales de un diagrama ER?

Los tres componentes de un diagrama entidad relación son:

Entidades

Las entidades, que suelen representarse en forma de rectángulo, son sustantivos (como objetos, personas, conceptos o eventos). Diferentes símbolos suelen reflejar distintos tipos de entidades. Por ejemplo, un conjunto de entidades fuerte suele representarse mediante un rectángulo, mientras que un conjunto de entidades débil se ilustra como un rectángulo doble.

Atributos

Los atributos son las características de una entidad. Por ejemplo, supongamos que su diagrama ER representa el software que utilizan los estudiantes para solicitar plaza en la universidad. Bajo la entidad "Estudiante", la siguiente información se mencionará como atributos: Número\_de\_estudiante, Nombre\_de\_estudiante y Curso\_de\_estudio.

Relaciones

En un diagrama ER, las líneas y flechas de conexión tienen etiquetas. Estas etiquetas representan cómo interactúan las entidades, mostrando al desarrollador las relaciones entre ellas. Los diferentes tipos de líneas de conexión también tienen diferentes significados y mostrarán a los desarrolladores cuántos datos fluyen de una entidad a la siguiente. Se trata de relaciones cardinales, que se explicarán con más detalle más adelante. Con estos tres componentes, los desarrolladores pueden mapear los elementos clave de una base de datos nueva o existente.

Ventajas de un diagrama entidad relación

Echa un vistazo a algunas de las principales ventajas de utilizar un diagrama entidad relación.

1. Visualiza las relaciones

Un ERD es un modelo físico de las relaciones entre entidades. Te permite visualizar cómo se relacionan las entidades entre sí, cómo funcionan sus relaciones y dónde se pueden mejorar estas relaciones.

2. Simplifica sistemas complejos

Desglosa sistemas y bases de datos complejos con un diagrama entidad relación. Mira cómo las entidades interconectadas se conectan y se superponen para comprender mejor cómo funciona el sistema.

3. Educa a empleados y partes interesadas

Utiliza un modelo lógico de datos para mostrar a los empleados nuevos y existentes cómo funcionan tus sistemas. Utilízalo para poner al día a las partes interesadas internas y externas y mantener a todo el mundo al tanto.

4. Es fácil de crear

Los modelos entidad relación son diagramas sencillos, es decir, fáciles de entender. No es necesario ser un desarrollador de software para crear el diagrama, ni se necesita ningún conocimiento técnico para entenderlo. Y con un espacio de trabajo virtual como Miro, puedes utilizar una plantilla de diagrama entidad relación ya preparada para empezar.

*Diagrama entidad relación: ¿Qué es para qué sirve? Con ejemplos | Miro*. (s. f.). https://miro.com/. https://miro.com/es/diagrama/que-es-diagrama-entidad-relacion/

**¿Qué es una base de datos relacional?**

Una base de datos relacional es una colección de información que organiza datos en relaciones predefinidas, en la que los datos se almacenan en una o más tablas (o "relaciones") de columnas y filas, lo que facilita la visualización y la comprensión de cómo se relacionan las diferentes estructuras de datos entre sí. Las relaciones son conexiones lógicas entre las diferentes tablas y se establecen a partir de la interacción entre ellas.

Descubre cómo las bases de datos relacionales de Google Cloud [Cloud SQL,](https://cloud.google.com/sql" \t "_blank) [Cloud Spanner](https://cloud.google.com/spanner)y [AlloyDB para PostgreSQL](https://cloud.google.com/alloydb" \t "_blank) pueden ayudarte a reducir los costos operativos y crear aplicaciones transformadoras.

¿Todo listo para comenzar? Crea una [instancia de prueba gratuita de Cloud Spanner](https://console.cloud.google.com/freetrial?redirectPath=/spanner) durante 90 días con 10 GB de almacenamiento sin costo.

Definición de una base de datos relacional

Una base de datos relacional (RDB) es una forma de estructurar información en tablas, filas y columnas. Un RDB tiene la capacidad de establecer vínculos (o relaciones) entre información mediante la unión de tablas, lo que facilita la comprensión y la obtención de estadísticas sobre la relación entre varios datos.

El modelo de base de datos relacional

Desarrollado por EF Codd desde IBM en la década de 1970, el modelo de base de datos relacional permite que cualquier tabla se relacione con otra mediante un atributo común. En lugar de usar estructuras jerárquicas para organizar los datos, Codd propuso un cambio a un modelo de datos en el que estos últimos se almacenan, se consultan y se relacionan en tablas sin reorganizar las tablas que los contienen.

Considera la base de datos relacional como una colección de archivos de hojas de cálculo que ayudan a las empresas a organizar, administrar y relacionar datos. En el modelo de base de datos relacional, cada “hoja de cálculo” es una tabla que almacena información, representada como columnas (atributos) y filas (registros o *tuplas*).

Los atributos (columnas) especifican un tipo de datos, y cada registro (o fila) contiene el valor de ese tipo de datos específico. Todas las tablas de una base de datos relacional tienen un atributo conocido como **clave primaria**, que es un identificador único de una fila, y cada fila se puede usar para crear una relación entre tablas diferentes mediante una **clave externa** (una referencia a una clave primaria de otra tabla existente).

Veamos cómo funciona el modelo de base de datos relacional en la práctica:

Supongamos que tienes una tabla de **clientes** y una de **pedidos**.

La tabla **Cliente** contiene datos sobre el cliente:

* ID de cliente (clave primaria)
* Nombre del cliente
* Dirección de facturación
* Dirección de envío

En la tabla de **clientes**, el ID del cliente es una clave primaria que identifica de forma única a ese cliente en la base de datos relacional. Ningún otro cliente tendría el mismo ID de cliente.

La tabla **Pedido** contiene información transaccional sobre un pedido:

* ID de pedido (clave primaria)
* ID de cliente (clave externa)
* Fecha del pedido
* Fecha de envío
* Estado del pedido

Aquí, la clave primaria para identificar un pedido específico es el ID de pedido. Puedes conectar un cliente con un pedido mediante el uso de una clave externa para vincular el ID de cliente de la tabla de **clientes**.

Las dos tablas ahora están relacionadas según el ID de cliente compartido, lo que significa que puedes consultar ambas tablas para crear informes formales o usar los datos en otras aplicaciones. Por ejemplo, un administrador de sucursal de venta minorista podría generar un informe sobre todos los clientes que realizaron una compra en una fecha específica o averiguar qué clientes tenían pedidos con una fecha de entrega retrasada en el último mes.

La explicación anterior debe ser simple. Sin embargo, las bases de datos relacionales también se destacan por mostrar relaciones muy complejas entre los datos, lo que permite hacer referencia a los datos en más tablas, siempre y cuando los datos cumplan con el esquema relacional predefinido de la base de datos.

Dado que los datos se organizan como relaciones predefinidas, puedes consultarlos de forma declarativa. Una consulta declarativa es una forma de definir lo que deseas extraer del sistema sin expresar cómo debería calcularse el resultado. Esto es fundamental para un sistema relacional, en lugar de otros sistemas.

Ejemplos de bases de datos relacionales

Ahora que comprendes cómo funcionan las bases de datos relacionales, puedes comenzar a conocer los diversos sistemas de administración de bases de datos relacionales que usa el modelo de base de datos relacional. Un sistema de administración de bases de datos relacionales (RDBMS) es un programa que se usa para crear, actualizar y administrar bases de datos relacionales. Algunos de los RDBMS más conocidos son MySQL, PostgreSQL, MariaDB, Microsoft SQL Server y Oracle Database.

Las bases de datos relacionales basadas en la nube, como [Cloud SQL](https://cloud.google.com/sql), [Cloud Spanner](https://cloud.google.com/spanner) y [AlloyDB](https://cloud.google.com/alloydb" \t "_blank) se volvieron cada vez más populares, ya que ofrecen servicios administrados para el mantenimiento de las bases de datos, parches, administración de capacidades, aprovisionamiento y asistencia de infraestructura.

¿Todo listo para comenzar? Crea una [instancia de prueba gratuita de Cloud Spanner](https://console.cloud.google.com/freetrial?redirectPath=/spanner) durante 90 días con 10 GB de almacenamiento sin costo.

Beneficios de las bases de datos relacionales

El principal beneficio del modelo de base de datos relacional es que proporciona una manera intuitiva de representar datos y permite un acceso fácil a datos relacionados. Por lo tanto, el uso de bases de datos relacionales es más común en las organizaciones que necesitan administrar grandes cantidades de datos estructurados, desde el seguimiento del inventario hasta el procesamiento de datos transaccionales y el registro de aplicaciones.

Existen muchas otras ventajas de usar bases de datos relacionales para administrar y almacenar tus datos:

Flexibilidad

Es fácil agregar, actualizar o borrar tablas, relaciones y hacer otros cambios a los datos cuando lo necesites sin cambiar la estructura general de la base de datos ni afectar las aplicaciones existentes.

Cumplimiento de ACID

Las bases de datos relacionales admiten el cumplimiento de ACID (atomicidad, coherencia, aislamiento y durabilidad) para garantizar la validez de los datos, sin importar si se producen errores, fallas o contratiempos de otro tipo.

Facilidad de uso

Es fácil ejecutar consultas complejas con SQL, lo que permite que incluso los usuarios no técnicos aprendan a interactuar con la base de datos.

Colaboración

Varias personas pueden operar y acceder a datos de manera simultánea. El bloqueo integrado impide el acceso simultáneo a los datos cuando se actualizan.

Seguridad integrada

La seguridad basada en roles garantiza que el acceso a los datos esté limitado a usuarios específicos.

Normalización de bases de datos

Las bases de datos relacionales usan una técnica de diseño conocida como normalización que reduce la redundancia de los datos y mejora su integridad.

Bases de datos relacionales y no relacionales

La principal diferencia entre las bases de datos relacionales y no relacionales (bases de datos NoSQL) es la forma en que se almacenan y organizan los datos. Las bases de datos no relacionales no almacenan datos de manera tabular y basada en reglas. En su lugar, almacenan datos como archivos individuales y no conectados, y se pueden usar para tipos de datos complejos y no estructurados, como documentos o archivos de rich media.

A diferencia de las bases de datos relacionales, las bases de datos NoSQL siguen un modelo de datos flexible, por lo que son ideales para almacenar datos que cambian con frecuencia o para aplicaciones que manejan tipos de datos diversos.

*¿Qué es una base de datos relacional (RDBMS)? | Google Cloud*. (s. f.). Google Cloud. https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419

**¿Qué es la normalización de bases de datos?**

Es el proceso de organizar los datos de una base de datos, valga la redundancia. Debemos tener en cuenta la creación de tablas y las reglas que se usan para definir las relaciones, estas reglas son diseñadas para proteger los datos, y para que la base de datos sea flexible con el fin de eliminar redundancias y dependencias incoherentes.

**¿Por qué se normalizan las bases de datos?**

Las bases de datos relacionales se normalizan para:

* Evitar la redundancia de los datos.
* Disminuir problemas de actualización de los datos en las tablas.
* Proteger la integridad de los datos.
* Facilitar el acceso e interpretación de los datos.
* Reducir el tiempo y complejidad de revisión de las bases de datos.
* Optimizar el espacio de almacenamiento.
* Prevenir borrados indeseados de datos.

**Requisitos de la normalización**

Para que las tablas de nuestra BD estén normalizadas deben cumplir las siguientes reglas:

* Cada tabla debe tener su nombre único.
* No puede haber dos filas iguales.
* No se permiten los duplicados.
* Todos los datos en una columna deben ser del mismo tipo.

**Reglas o niveles de normalización**

Para normalizar una base de datos existen principalmente 3 reglas, las cuales se deberían cumplir para evitar redundancias e incoherencias en las dependencias. A estas reglas se les conoce como **"Forma normal"** qué va de la 1 a la 3 y si la base de datos cumple con cada regla se dice que está en la **"primera o segunda o tercera forma normal"**

*Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayoría de las aplicaciones.*

**Primera forma normal**

* Elimine los grupos repetidos de las tablas individuales.
* Cree una tabla independiente para cada conjunto de datos relacionados.
* Identifique cada conjunto de datos relacionados con una clave principal.

**Segunda forma normal**

* Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.
* Relacione estas tablas con una clave externa.

**Tercera forma normal**

* Elimine los campos que no dependan de la clave.

Cada nivel de normalización exige cierta complejidad, pero debes analizar si en realidad necesitas llegar hasta la tercera forma normal, debes buscar lo que mejor se adapte a la empresa.

*Normalización de bases de datos*. (s. f.). EDteam - En Español Nadie Te Explica Mejor. https://ed.team/blog/normalizacion-de-bases-de-datos

Qué es Java?

Java es un lenguaje de programación multiplataforma orientado a objetos que se ejecuta en miles de millones de dispositivos de todo el mundo. Sustenta aplicaciones, sistemas operativos de smartphones, software empresarial y muchos programas conocidos. A pesar de haberse inventado hace más de 20 años, Java es actualmente el lenguaje de programación más popular para los desarrolladores de aplicaciones.

Estos son los rasgos que sirvieron para definir el lenguaje de programación Java y que lo hicieron tan ubicuo. Java es:

**Multiplataforma:** Java se catalogó con el eslogan "Escribir una vez, ejecutar en cualquier lugar" (o WORA, por sus siglas en inglés), lo cual sigue siendo cierto hoy en día. El código de programación Java escrito para una plataforma (como el sistema operativo Windows) puede transferirse fácilmente a otra plataforma (como el sistema operativo de un teléfono móvil) y viceversa, sin necesidad de reescribirlo por completo. Java funciona en múltiples plataformas porque, al compilar un programa java, el compilador crea un archivo de código de bytes .class que se puede ejecutar en cualquier sistema operativo que tenga instalada la máquina virtual Java (JVM). Normalmente es fácil instalar JVM en la mayoría de los sistemas operativos principales, incluido iOS, lo cual no siempre ha sido así.

**Orientado a objetos:** Java se encontraba entre los primeros lenguajes de programación orientados a objetos. Un lenguaje de programación orientado a objetos organiza su código en torno a clases y objetos, en lugar de funciones y comandos. La mayoría de los lenguajes de programación modernos, incluidos C++, C#, Python y Ruby, están orientados a objetos.

Estos rasgos hacen que Java sea un lenguaje de programación versátil y relativamente fácil de aprender que es muy popular entre las empresas de software y tecnología. Si quiere hacer carrera en codificación o programación informática, saber cómo escribir Java es una habilidad valiosa.

¿Para qué se usa el lenguaje de programación Java?

Java es un lenguaje de programación muy transferible que se usa en plataformas y distintos tipos de dispositivos, desde smartphones hasta televisores inteligentes. Se usa para crear aplicaciones móviles y web, software empresarial, dispositivos de Internet de las cosas (IoT), juegos, macrodatos, aplicaciones distribuidas y basadas en la nube, entre otros tipos. Estos son algunos ejemplos específicos del mundo real de las aplicaciones que se programan con Java.

**Aplicaciones móviles**

Muchas aplicaciones móviles, si no la mayoría, se compilan con Java. Java es uno de los lenguajes preferidos de los desarrolladores de aplicaciones móviles debido a su plataforma estable y su versatilidad. Entre las aplicaciones móviles populares codificadas en Java se incluyen Spotify, Signal y Cash App.

**Aplicaciones web**

Una amplia variedad de aplicaciones web se desarrollan con Java. Twitter y LinkedIn se encuentran entre las más conocidas.

**Software empresarial**

El software empresarial está pensado para servir a un grupo o una organización de gran tamaño. Incluye software como sistemas de facturación y programas de administración de cadenas de suministros. La alta escalabilidad de Java lo convierte en un lenguaje atractivo para los desarrolladores que escriben software empresarial.

**Juegos**

Entre los juegos populares escritos en el lenguaje de programación Java se incluyen los originales Minecraft y RuneScape.

**Aplicaciones de IoT**

Las aplicaciones de IoT están en todas partes (televisores inteligentes, coches, maquinaria pesada, instalaciones de trabajo, etc.) y Java se usa para programar muchas de ellas. Java es una opción popular para los desarrolladores de IoT por la facilidad con la que puede transferirse su código entre plataformas.

¿Cómo funciona Java?

Como se ha explicado anteriormente, Java es un lenguaje de programación multiplataforma. Esto significa que se puede escribir para un sistema operativo y ejecutarse en otro. ¿Cómo es posible?

El código Java se escribe primero en un kit de desarrollo de Java, que están disponibles para Windows, Linux y macOS. Los programadores escriben en el lenguaje de programación Java, que el kit traduce a código informático que cualquier dispositivo puede leer con el software adecuado. Esto se consigue con un programa llamado compilador. Un compilador toma el código informático de alto nivel, como Java, y lo traduce a un lenguaje que los sistemas operativos entienden, llamado código de bytes.

A continuación, un intérprete llamado Máquina virtual Java (JVM) procesa el código de bytes. Las JMV están disponibles para la mayoría de las plataformas de software y hardware, y esto es lo que permite transferir código Java de un dispositivo a otro. Para ejecutar Java, las JVM cargan el código, lo comprueban y proporcionan un entorno en tiempo de ejecución.

Dada la alta portabilidad de Java, no es ninguna sorpresa que muchas personas quieran aprender a escribir en este lenguaje. Afortunadamente, hay muchos recursos disponibles para empezar a aprender Java.

*¿Qué es Java?* (s/f). Microsoft.com. Recuperado el 8 de abril de 2025, de <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-java-programming-language>

Un **IDE (Integrated Development Environment)** o entorno de desarrollo integrado es un entorno digital que permite el desarrollo de programas, videojuegos o cualquier otro elemento relacionado con código.

Los IDE facilitan el trabajo a desarrolladores de software, así como a programadores informáticos.

Gracias a los IDE, los pasos básicos de un [proceso de desarrollo de software](https://immune.institute/proceso-desarrollo-software-ciclo-vida/) quedan integrados, así como los de la escritura del código, entre otras acciones.

En este post, hablamos sobre un IDE muy utilizado en la actualidad: **NetBeans, el cual ha sido creado, sobre todo, para trabajar con el lenguaje de programación Java.**

Netbeans o Apache NetBeans: definición

**¿Qué es NetBeans?** Antes de nada, cabe destacar que a pesar de que todo el mundo conoce a NetBeans como “NetBeans”, tras un proceso de rebranding en 2016, se cambió el nombre a “Apache NetBeans”. De ahí el título de este apartado. Ahora sí, es el momento de conocer su definición:

NetBeans es un IDE o entorno de desarrollo integrado, basado en el lenguaje Java y ejecutado en Swing.

De esta forma, NetBeans o Apache NetBeans es una aplicación de código abierto, que ha cobrado bastante popularidad en los últimos años.

Este IDE**, orientado principalmente a las apps de Java**, ofrece diferentes herramientas digitales como editor de texto, código, compilador, interfaz gráfica de usuario; además de un depurador.

Por otro lado, cabe destacar que NetBeans facilita la creación de aplicaciones estructuradas, ya que están basadas en un conjunto de módulos. Así, se favorece el desarrollo de las diversas funciones de una manera independiente y pudiendo también reutilizar los componentes.

NetBeans y Java

Si quieres responder de verdad qué es NetBeans y para qué sirve, **hay que situar esta aplicación junto al lenguaje de Java**. Y es que ambos pertenecían a la misma compañía, de ahí que este IDE se encuentre orientado principalmente a dicho lenguaje de programación.

No obstante, NetBeans se puede utilizar sin ningún inconveniente con otros [lenguajes de programación](https://immune.institute/lenguajes-de-programacion-para-movil/) para la creación y desarrollo de programas informáticos.

Por consiguiente, podemos crear y desarrollar aplicaciones con **NetBeans y PHP**o, incluso, con lenguajes más dinámicos como es el caso de JavaScript. A ello se le suma una gran lista de otros lenguajes, tales como: [C](https://immune.institute/lenguaje-de-programacion-c/), Ruby, C++ o Groovy, entre otros.

¿Para qué sirve NetBeans?

**NetBeans ayuda en el desarrollo de cualquier tipo de software y en sus distintas fases.** A los programadores les sirve como medio para la escritura, compilación o, simplemente, para hallar errores en programas basados en los anteriores lenguajes.

Es útil para ejecutar programas en diferentes dispositivos electrónicos, tanto móviles como PC. Asimismo, NetBeans permite a través de Bookmarking marcar las líneas de código que a posteriori se modificarán.

Por otro lado, proporciona una librería visual donde conseguir los diferentes widgets y permite **gestionar el almacenamiento de los datos de una forma óptima**.

Ventajas de Netbeans

Para entender qué es NetBeans, es importante conocer **cuáles son los beneficios de utilizar este IDE**. Veamos a continuación algunos ejemplos:

* **Código abierto:** NetBeans es de código abierto y por ende gratuito, de modo que cualquiera puede descargarse esta aplicación. Para ello, es necesario visitar la web oficial de [Apache NetBeans](https://netbeans.apache.org/download/index.html).
* **Multiplataforma:**Se puede emplear en distintos dispositivos, así como ejecutarse en diferentes sistemas operativos (Mac OS, Windows, Linux y Solaris).
* **Manejo automático de la memoria:**Para aquellos programas con C o C++, la administración de la memoria se puede realizar de forma automática.
* **Multilenguaje:** Como venimos diciendo, NetBeans permite desarrollar aplicaciones multilenguaje. Es decir, no solo opera con Java; sino que NetBeans puede utilizar otros lenguajes como PHP, C o Ruby, entre otros.

López, M. (2022, 13 abril). ¿Qué es NetBeans? Ventajas y usos - IMMUNE Institute. *IMMUNE Technology Institute*. https://immune.institute/blog/que-es-netbeans/