

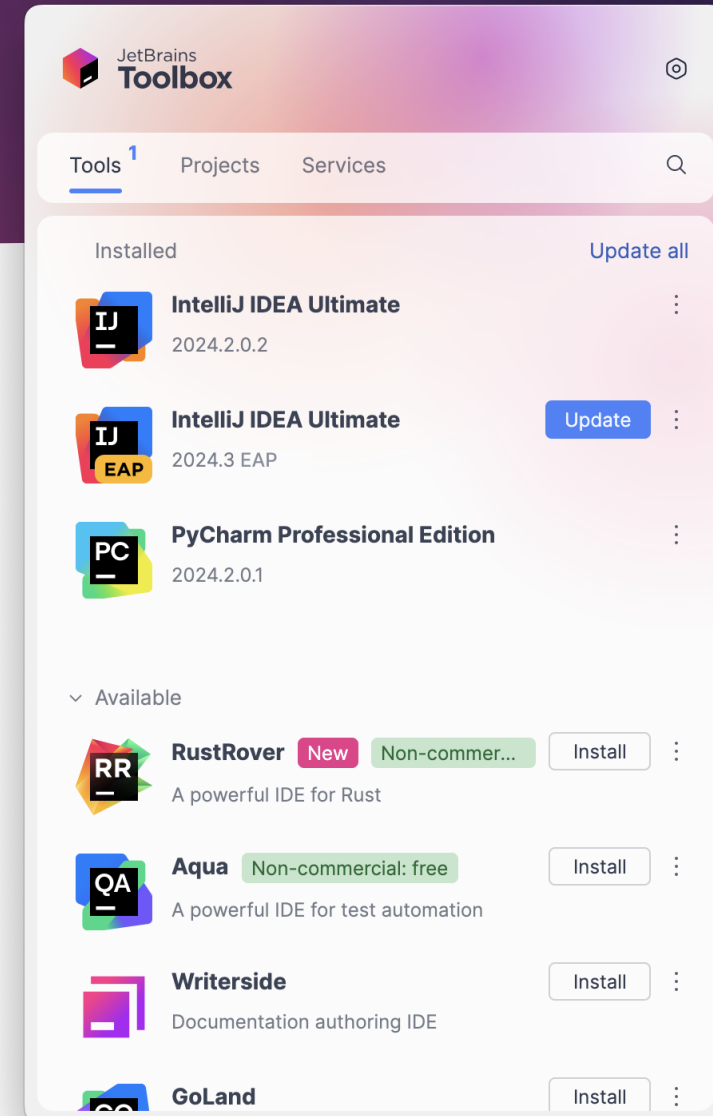


# Mejora tu productividad con IntelliJ

JUAN JOSÉ GONZÁLEZ ABRIL

# JetBrains Toolbox

- ▶ Actualizaciones rápidas y sencillas
- ▶ Consultar novedades
- ▶ Mantener varias versiones en paralelo
- ▶ Abrir proyectos al instante



# Versiones de IntelliJ

- ▶ Community
  - ▶ Gratuita
  - ▶ Open-source
  - ▶ Perfecta para trabajar con Java y Kotlin
- ▶ Ultimate \$
  - ▶ De pago
  - ▶ Soporte para frameworks web (Spring, Java EE, Jakarta EE)
  - ▶ Herramientas de bases de datos

# Filosofía de IntelliJ

- ▶ Diseñado para usarse principalmente con el teclado
- ▶ Existen atajos de teclado para casi todo
  - ▶ [Help | Keyboard Shortcuts PDF](#)
- ▶ Seguimiento de lo productivo que eres
  - ▶ [Help | My Productivity](#)

# Atajos generales

- ▶ Configuración rápida: ^` / Ctrl+`
- ▶ Ajustes: ⌘, / Ctrl+Alt+S
- ▶ Estructura del proyecto: ⌘; / Ctrl+Alt+Shift+S

# Atajos esenciales

- ▶ “Show intention actions and quick-fixes”:  $\text{⇧⇧}$  / Alt+Enter
- ▶ “Search Everywhere”:  $\text{⇧⇧}$  / Doble Shift

# Navegación: General

- ▶ Tool windows: ⌘0 ... ⌘9 / Alt+0 ... Alt+9
- ▶ Archivos recientes: ⌘E / Ctrl+E
  - ▶ Ubicaciones recientes: ⌘⇧E / Ctrl+Shift+E
- ▶ Navegar hacia atrás: ⌘⇐ / Ctrl+Alt+⇐
- ▶ Navegar hacia adelante: ⌘⇒ / Ctrl+Alt+⇒
- ▶ Seleccionar fichero actual en cualquier vista: ⌘F1

# Navegación: Ficheros

- ▶ Ir a línea: ⌘L / Ctrl+G
- ▶ Ir a declaración: ⌘B / Ctrl+B
  - ▶ Ir a implementación: ⇧⌘B / Ctrl+Alt+B
  - ▶ Ir a declaración del tipo: ⌘⇧B / Ctrl+Shift+B
- ▶ Ir a *super*-método / *super*-clase: ⌘U / Ctrl+U
- ▶ Método anterior: ^⌘↑ / Alt+↑
- ▶ Método siguiente: ^⌘↓ / Alt+↓
- ▶ Ver fuente: ⌘↵ / Ctrl+Enter



# Navegación: Errores

- ▶ Siguiente error: F2 / F2
- ▶ Error anterior: ↑F2 / Shift+F2

# Generación de Código

- ▶ “Generate code”: ⌘N / Alt+Ins
  - ▶ Contstructores
  - ▶ Getters y setters
  - ▶ Métodos equals, hashCode y toString
  - ▶ Implementar y sobrescribir métodos
- ▶ “Postfix completion”
  - ▶ Permite añadir código a partir de una plantilla alrededor de la expresión que acabas de escribir
- ▶ “Live templates”
  - ▶ Permite insertar estructuras comunes, como bucles, condiciones, declaraciones o *print statements*.

# Refactor

- ▶ “Refactor This”: ^T / Ctrl+Alt+Shift+T
  - ▶ Renombrar y cambiar firma
  - ▶ Introducir variables, parámetros, etc
  - ▶ Extraer/“Inline” métodos
  - ▶ Extraer interfaz, superclase, etc
  - ▶ “Pull up”/“Push down” de métodos
  - ▶ Mover métodos y clases

# Estilo de Código

- ▶ Formateo del código de acuerdo con reglas predefinidas
- ▶ Permite exportar configuraciones
- ▶ Soporte para .editorconfig

# Inspecciones

- ▶ Validaciones sobre la estructura de código
- ▶ Diferentes convenciones de programación
- ▶ Estandariza el código
- ▶ Puede incluirse en un repositorio

# “Run configurations”

- ▶ Configuraciones que definen:
  - ▶ que ejecutar
  - ▶ con qué parámetros
  - ▶ con que entornos
- ▶ Se pueden crear dependencias entre configuraciones
- ▶ Pueden incluirse en un repositorio

# Bases de datos

- ▶ Cliente BBDD \$
  - ▶ Autocompletado
  - ▶ Modificación de tablas mediante UI
- ▶ JPA Buddy \$
  - ▶ Generación de entidades a partir de tablas y viceversa
  - ▶ Modificación de entidades mediante UI
  - ▶ Validación entidades de Hibernate

# HTTP

- ▶ Cliente HTTP completo \$
  - ▶ Variables de entorno
  - ▶ Múltiples entornos
  - ▶ Scripting usando JavaScript
  - ▶ Autocompletado en basado endpoints existentes



# Local History

- ▶ Historial local de todos los cambios
- ▶ Se mantienen entre reinicios
- ▶ “Te puede salvar la vida”

# Cliente de Git

- ▶ Gestión completa desde la UI de IntelliJ
- ▶ Resolución de conflictos de forma ágil
- ▶ “Shelve” independiente del “Stash”
- ▶ Validaciones antes de hacer commit
- ▶ Revisión de pull requests
  - ▶ GitHub
  - ▶ GitLab