

## Detalhes das Plataformas AWS AI

[Comandos](#) • [Funcionalidades](#) • [Integrações](#) | Guia Completo

### Amazon Q Developer IntelliJ (AWS Toolkit)

Versão: 3.99 (21/11/2025)

IDEs Suportadas: Todos os JetBrains IDEs 2023.3+

*IntelliJ IDEA, PyCharm, WebStorm, PhpStorm, GoLand, Rider, CLion, RubyMine, DataGrip, AppCode*

#### Comandos de Chat:

- `/clear` - Limpar conversa atual
- `/compact` - Compactar histórico de chat
- `/help` - Exibir ajuda e comandos disponíveis
- `/model` - Selecionar modelo AI
- `/dev` - ⚠️ Depreciado v3.85 (use conversação natural)
- `/test` - ⚠️ Depreciado v3.85 (use conversação natural)
- `/doc` - ⚠️ Depreciado v3.85 (use conversação natural)
- `/review` - ⚠️ Depreciado v3.68 (use conversação natural)

#### Funcionalidades Principais:

- ✓ Sugestões inline em tempo real (auto-trigger)
- ✓ Chat contextual sobre código
- ✓ Agente coding (atualiza arquivos diretamente)
- ✓ Desenvolvimento de features (conversação natural)
- ✓ Transformação e modernização de código
- ✓ Personalização com modelos personalizados
- ✓ Project Rules (`amazonq/rules/`)
- ✓ MCP Server Management

#### Serviços AWS Integrados:

- CloudFormation: View events, resources, outputs
- CloudWatch Logs: View e search de log streams
- Lambda: Run/Debug local, invoke remote, package & deploy
- RDS/Redshift: Conexão com credenciais temporárias
- S3: View, manage, upload/download buckets
- SAM: Sync & track SAM-based applications

#### Linguagens Suportadas (SAM):

Java, Python, Node.js, .NET Core, Go

### Amazon Q Developer Visual Studio Code

Versão: 3.89.0 (25/11/2025)

IDE: Visual Studio Code

#### Comandos de Chat:

- `/clear` - Limpar conversa atual
- `/compact` - Compactar histórico de chat
- `/help` - Exibir ajuda e comandos disponíveis
- `/model` - Selecionar modelo AI
- `/dev, /test, /doc` - ⚠️ Deprecados v3.85
- `/review` - ⚠️ Deprecado v3.68

#### Funcionalidades Principais:

- ✓ Sugestões inline em tempo real
- ✓ Inline chat (Ctrl+I / Cmd+I)
- ✓ Chat contextual sobre código
- ✓ Agente coding (atualiza arquivos)
- ✓ Project Rules (`amazonq/rules/`)
- ✓ MCP Server Management
- ✓ Context management (@workspace)

#### Serviços AWS Integrados:

- CloudFormation, Lambda, S3
- SAM, CloudWatch Logs, ECS
- DynamoDB Table viewer

#### Linguagens Suportadas:

Java, Python, JavaScript, TypeScript, C#, Go, Ruby, PHP, Kotlin, Rust, Scala, Shell, SQL, C, C++

### Kiro CLI (ex-Amazon Q CLI)

Versão: v1.19.7 → Kiro CLI

Ambientes: macOS, Linux (AppImage, Ubuntu), AWS CloudShell

⚠️ IMPORTANTE: Amazon Q CLI foi renomeado para Kiro CLI em 17/11/2025. Comandos q e q\_chat continuam funcionando (retrocompatibilidade).

#### Comandos Principais:

- `kiro-cli chat ou q_chat` - Iniciar sessão de chat interativa
- `q translate` - Traduzir linguagem natural para comandos shell
- `q inline enable/disable` - Gerenciar sugestões inline
- `q doctor` - Diagnosticar problemas
- `q update` - Atualizar CLI
- `q login` - Autenticar (Builder ID, IAM, GitHub, Gmail)

#### Comandos dentro do Chat (kiro-cli chat):

- `/clear, /compact, /help, /usage`
- `/editor` - Abrir editor para prompts longos
- `/model` - Selecionar modelo (Claude Sonnet 4, 3.7, 3.5)
- `/save, /load` - Exportar/importar conversa JSON
- `/tools` - Gerenciar ferramentas disponíveis
- `/mcp` - Gerenciar servidores MCP
- `/agent` - Gerenciar agentes personalizados
- `/knowledge` - Gerenciar base de conhecimento (PDFs)
- `/tangent` - Criar checkpoints de conversa
- `!` - Executar comando shell diretamente

#### Funcionalidades Principais:

- ✓ Autocompletar para CLIs populares (git, npm, docker, aws, kubectl)
- ✓ Tradução de linguagem natural para comandos shell
- ✓ Custom agents (agentes personalizados)
- ✓ Enhanced thinking mode (raciocínio estendido)
- ✓ Knowledge base (PDFs, documentos)
- ✓ Tangent conversations (checkpoints)
- ✓ MCP Server support
- ✓ Suporte a imagens (paste)

#### Autenticação Suportada:

- AWS Builder ID
- IAM Identity Center (SSO)
- GitHub (Kiro CLI)
- Gmail (Kiro CLI)

## Kiro IDE

Status: General Availability (GA) desde 17/11/2025

Base: VS Code (Code-OSS)

Modelos AI: Claude Sonnet 4 (principal), Claude 3.7 Sonnet (fallback)

### Workflow Spec-Driven Development:

1. **Prompt** → Descreva o que quer construir em linguagem natural
2. **Spec** → Kiro transforma em especificações técnicas detalhadas
3. **Code** → Gera código funcional, documentação e testes automaticamente

### Funcionalidades Principais:

- Desenvolvimento spec-driven (diferencial único)
- Agentes integrados para problemas complexos
- Geração automática de documentação e testes
- Código production-ready desde o início
- Checkpointing (rewind de modificações)
- Property-based tests
- Custom agents e hooks
- MCP (Model Context Protocol) support

### Planos de Assinatura:

- **Free:** Uso básico com limitações
- **Pro:** Recursos avançados
- **Pro+:** Recursos premium
- **Power:** Uso ilimitado para equipes

### Diferenciais vs Outras IDEs:

- Workflow completo: Prompt → Spec → Code
- Código production-ready desde o início
- Especificações técnicas detalhadas antes do código
- Validação automática contra especificações
- Property-based testing integrado

### Idiomas Suportados (Chat)

Todos os ambientes suportam chat em **10 idiomas**:

- Inglês
- Mandarim
- Francês
- Alemão
- Italiano
- Japonês
- Espanhol
- Coreano
- Hindi
- Português

[Próximo: Exemplos de Prompts](#) → ← Anterior: Comparação de Plataformas

Data de Publicação: 27 de Novembro de 2025

Conteúdo educacional baseado em documentação pública. Não afiliado à AWS.