

Guía Completa de Migración Ruby 2 a Ruby 3

Documento generado: 31 de Julio de 2025

Versión: 1.0.0

Herramienta: Ruby Migrator Tool

Resumen Ejecutivo

Ruby 3 introduce cambios significativos que mejoran el rendimiento, la seguridad y la consistencia del lenguaje. Esta guía documenta todos los cambios críticos, advertencias y mejores prácticas para migrar código de Ruby 2.x a Ruby 3.x.

Cambios Críticos (Rompen Compatibilidad)

1. Clases Numéricas Unificadas

ELIMININADO: Las clases Fixnum y Bignum han sido removidas completamente.

Ruby 2 (Problemático):

```
if num.is_a?(Fixnum)
  puts "Es un número pequeño"
elsif num.is_a?(Bignum)
  puts "Es un número grande"
end
```

Ruby 3 (Correcto):

```
if num.is_a?(Integer)
  puts "Es un número entero"
end
```

2. Constantes Booleanas y Nil

ELIMINADO: Las constantes TRUE, FALSE y NIL han sido removidas.

Ruby 2 (Problemático):

```
def check_status
  return TRUE if valid?
  return FALSE if invalid?
  return NIL
end
```

Ruby 3 (Correcto):

```
def check_status
  return true if valid?
  return false if invalid?
  return nil
end
```

3. Métodos de Seguridad Obsoletos

ELIMINADO: Los métodos taint, untaint, trust y untrust han sido completamente removidos.

Ruby 2 (Problemático):

```
def process_input(data)
  data.taint
  clean_data = sanitize(data)
  clean_data.untaint
  data.trust
  return clean_data
end
```

Ruby 3 (Correcto):

```
def process_input(data)
  clean_data = sanitize(data)
  return clean_data
end
# Nota: Usar validación explícita en lugar de taint
```

⚠ Advertencias Importantes (Cambios de Comportamiento)

1. Variables Globales de Coincidencia

CAMBIO: Las variables \$1, \$2, etc. tienen comportamiento modificado en algunos contextos.

Ruby 2 (Problemático):

```
if url =~ /https?:\V(\w+):(\d+)/
  host = $1
  port = $2.to_i
end
```

Ruby 3 (Recomendado):

```
if match = url.match(/https?:\V(?:<host>\w+):(?:<port>\d+)/)
  host = match[:host]
  port = match[:port].to_i
end
```

2. Argumentos de Palabra Clave

CAMBIO: Separación estricta entre argumentos posicionales y de palabra clave.

Ruby 2 (Ambiguo):

```
def method_with_options(arg, options = {})
  # ...
end

method_with_options("value", key: "value")
```

Ruby 3 (Explícito):

```
def method_with_options(arg, **options)
  # ...
end

method_with_options("value", key: "value")
```

3. Sintaxis de Hash

OBSOLETO: La sintaxis de hash rockets está obsoleta para símbolos.

Ruby 2 (Obsoleto):

```
config = {
  :host => "localhost",
  :port => 3000
}
```

Ruby 3 (Moderno):

```
config = {  
  host: "localhost",  
  port: 3000  
}
```

Tabla de Compatibilidad Completa

Característica	Ruby 2	Ruby 3	Severidad	Acción Requerida
Fixnum/Bignum	Clases separadas	Solo Integer	ERROR	Reemplazar con Integer
TRUE/FALSE/NIL	Constantes disponibles	Removidas	ERROR	Usar true/false/nil
taint/untaint	Métodos disponibles	Removidos	ERROR	Remover llamadas
trust/untrust	Métodos disponibles	Removidos	ERROR	Remover llamadas
Variables \$1, \$2	Comportamiento global	Comportamiento local	ADVERTENCIA	Usar capturas nombradas
Hash rockets	:key => value	key: value (preferido)	ADVERTENCIA	Actualizar sintaxis
Argumentos palabra clave	Mezclados con posicionales	Separación estricta	INFO	Usar ** explícitamente
Rendimiento	Línea base	3x más rápido	INFO	Beneficio automático

Herramientas de Migración

Ruby Migrator Tool

Herramienta automatizada que analiza tu código y detecta problemas de compatibilidad:

```
# Análisis sin modificar archivos
ruby migrator.rb -p mi_proyecto -r

# Análisis con respaldos automáticos
ruby migrator.rb -p mi_proyecto

# Reporte en formato JSON
ruby migrator.rb -p mi_proyecto -r -f json
```

Características de la Herramienta

- ☐ Detección automática de 24+ patrones problemáticos
- ☐ Análisis AST (Abstract Syntax Tree) preciso
- ☐ Respaldos automáticos con timestamp
- ☐ Reportes detallados en texto y JSON
- ☐ Interfaz web interactiva
- ☐ Correcciones automáticas para casos simples

☐ Mejoras de Rendimiento en Ruby 3

Mejoras principales:

- **3x más rápido** en muchas operaciones
- **JIT (Just-In-Time) compilation** mejorado
- **Fiber Scheduler** para concurrencia
- **Ractor** para paralelismo real
- **Recolector de basura** optimizado

☐ Plan de Migración Paso a Paso

1. **Análisis inicial:** Ejecutar Ruby Migrator Tool en modo solo lectura
2. **Priorizar errores:** Corregir todos los problemas marcados como ERROR
3. **Revisar advertencias:** Evaluar y corregir problemas de ADVERTENCIA
4. **Actualizar dependencias:** Verificar compatibilidad de gemas
5. **Pruebas exhaustivas:** Ejecutar suite completa de tests
6. **Migración gradual:** Migrar por módulos o componentes
7. **Monitoreo post-migración:** Verificar rendimiento y errores

⚡ Ejemplos Completos de Migración

Clase Legacy Completa

Ruby 2 (Antes):

```
class LegacyProcessor
  def initialize(options = {})
    @config = { :timeout => 30, :retries => 3 }.merge(options)
  end

  def process(data)
    data.taint
    return FALSE unless validate(data)

    if data.is_a?(Fixnum) || data.is_a?(Bignum)
      result = calculate(data)
    else
      return NIL
    end

    result.untrust
    return TRUE
  end
end
```

Ruby 3 (Después):

```
class ModernProcessor
  def initialize(**options)
    @config = { timeout: 30, retries: 3 }.merge(options)
  end

  def process(data)
    return false unless validate(data)

    if data.is_a?(Integer)
      result = calculate(data)
    else
      return nil
    end

    return true
  end
end
```

❑ Lista de Verificación Final

Antes de considerar completa la migración:

- ☐ Todos los errores ERROR corregidos
- ☐ Advertencias WARNING revisadas

- ☐ Tests pasando en Ruby 3
- ☐ Dependencias actualizadas
- ☐ Rendimiento verificado
- ☐ Documentación actualizada
- ☐ Equipo capacitado en cambios

Soporte y Recursos

- **Ruby Migrator Tool:** Herramienta de análisis automático
- **Documentación oficial:** ruby-doc.org
- **Changelog Ruby 3:** Cambios detallados por versión
- **Comunidad Ruby:** Foros y grupos de ayuda

Documento generado automáticamente por Ruby Migrator Tool v1.0.0

Fecha: 31 de Julio de 2025

Esta guía proporciona información completa para migrar exitosamente de Ruby 2 a Ruby 3.