PLC Version Control

Project Name

`<ProjectID>` - `<vHw.F.Sw>` - `<yyyy-mm-dd>`

Example:
✅ *P123-v1.1.6-2025-11-05*

🎲 Starts at 1

🦺 0 to n, when Hw changes, F is set to 0

🔣 0 to n, when Hw or F changes, Sw is set to 0

🥙 Kebab-case
🅿️ Project ID
🎲 Hw: Hardware version
⛏️ F: Safety verion
🔣 Sw: Software version
📅 Fecha (ISO8601)

Code version

Implements Semantic Versioning (SemVer)

`vM.m.P`

🔼 M: Major version
👆 m: Minor version
🩹 P: Path/Hot fix

```
Example:
Function _foo() gets date as DTL and
return date as string in ISO format
```

Commit

📍 **fix**: Fix something
📍 **feat**: Feature
📍 **docs**: Documentation
📍 **style**: Format code
📍 **refactor**: Clean code, enhance code
📍 **perf**: Performance
📍 **test**: Testing
📍 **chore**: Update libraries, dependencies
📍 **build**: Compile, download

Commit	Change
feat: First version	v1.0.0
fix: Fix year	PATCH → v1.0.1
docs: Update readme	❌ no cambia
style: Format if clauses	❌ no cambia
refactor: Enhance code	MINOR → v1.1.0
perf: Improve DTL to string conversion	MINOR → v1.2.0
feat: Change DTL input to DateTime	MAJOR → v2.0.0 (breaking change)
test: Unitary test	❌ no cambia
build: Download to PLC	❌ no cambia