



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

# UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

## INGENIERÍA EN SOFTWARE

Trabajo de titulación previo a la obtención del título de Ingeniero en Software

***“DESARROLLO DE UN LENGUAJE DE DOMINIO ESPECÍFICO (DSL) PARA PROGRAMAR UNA PLACA ARDUINO UTILIZANDO EL PARADIGMA DE INGENIERÍA DIRIGIDA POR MODELOS (MDE)”***

Autores:

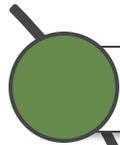
**Franco Román, Jonathan Rubén  
Sánchez Tapia, Benjamín Alejandro**

Ph.D. Jácome Guerrero Patricio Santiago, ***Director***

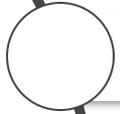


# Itinerario del Día

- **Introducción**
- **Planteamiento del problema**
- **Fundamentos teóricos**
- **Desarrollo del DSL**
- **Análisis de resultados**
- **Conclusiones y recomendaciones**



## Introducción



Planteamiento del problema



Fundamentos teóricos



Desarrollo del DSL



Análisis de resultados



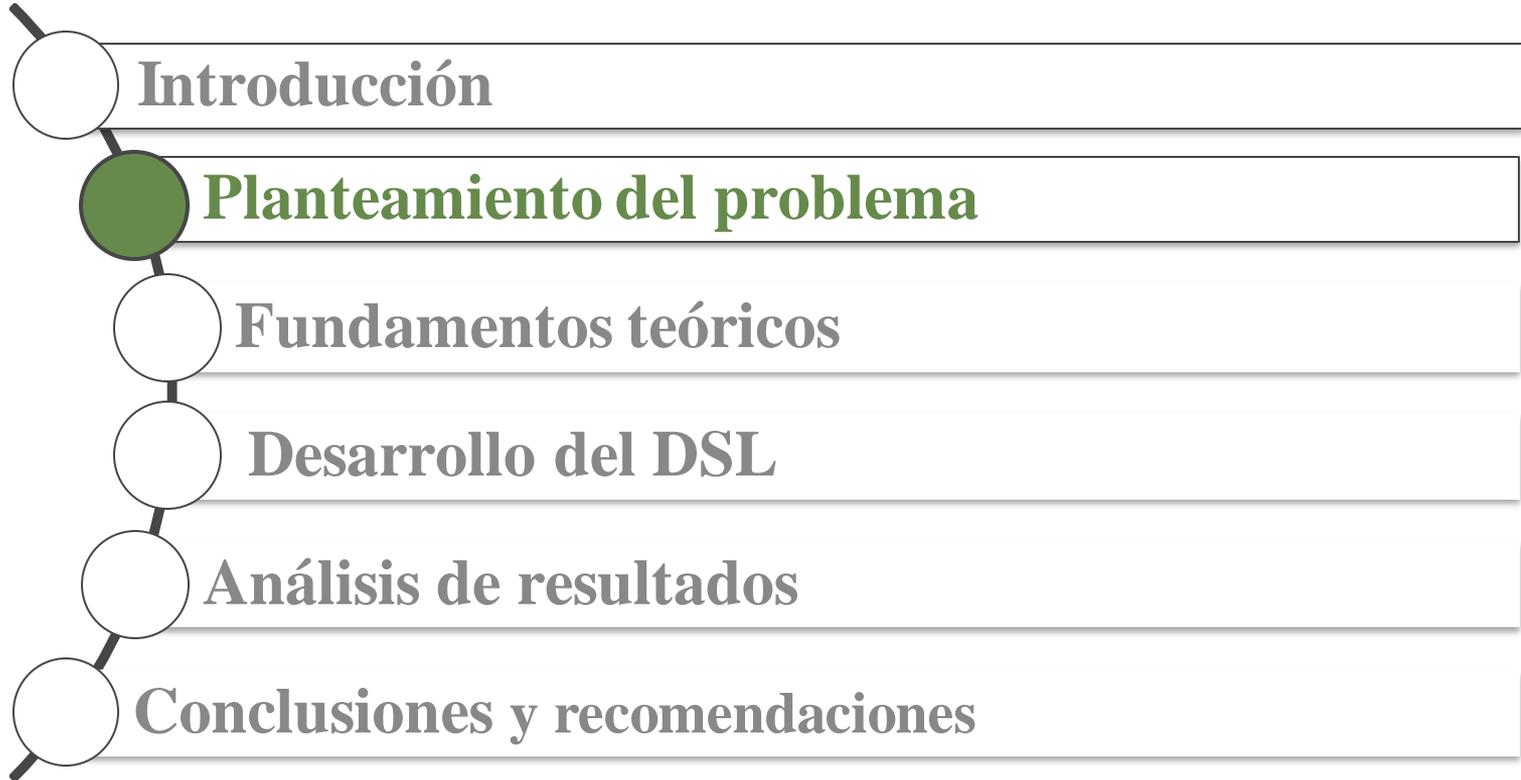
Conclusiones y recomendaciones



# Introducción



- **Desarrollo de un Lenguaje de Dominio Específico (DSL).**
- **Simular circuitos lógicos básicos con compuertas AND, NOT y OR.**
- **Automatizar el proceso de programación manual.**



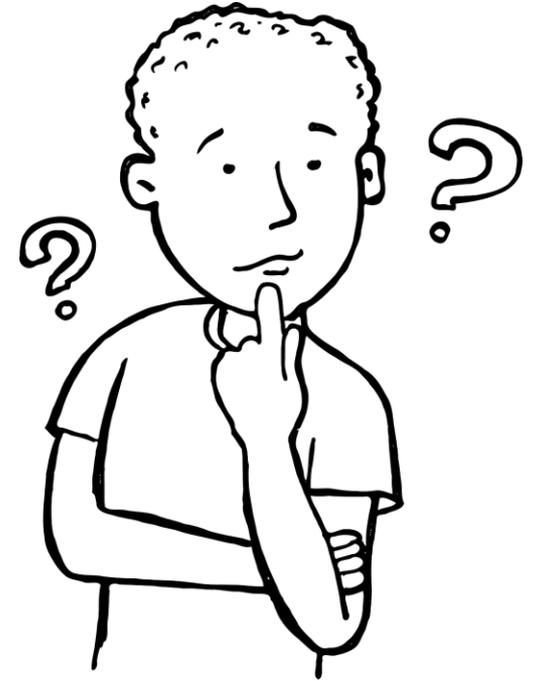
# Antecedentes



- ✓ Actualmente la producción de software en gran mayoría aún es desarrollada por procesos manuales.
- ✓ Industria 4.0.
- ✓ Ingeniería Dirigida por Modelos (MDE).

# Formulación del problema

¿Cómo desarrollar un DSL que permita optimizar el tiempo de programación de código funcional para la placa Arduino utilizando compuertas lógicas “AND”, “OR” y “NOT”?.



# Objetivos General

Desarrollar un DSL que permita optimizar el tiempo de programación de la placa Arduino UNO utilizando compuertas lógicas “AND”, “OR” y “NOT”.



# Objetivos Específicos



Formular el marco teórico para el desarrollo de un DSL que permita optimizar el tiempo de programación de la placa Arduino UNO utilizando compuertas lógicas “AND”, “OR” y “NOT”.



Desarrollar el DSL para optimizar el tiempo de programación de código funcional para la placa Arduino UNO utilizando compuertas lógicas “AND”, “OR” y “NOT”.



Definir la sintaxis abstracta (Metamodelo) del DSL para definir la estructura del dominio del lenguaje.

# Objetivos Específicos



Definir la sintaxis concreta (Representación gráfica) del DSL para especificar la notación específica con la que los usuarios interactuarán.



Desarrollar el generador de código a partir del DSL a código para programar la placa Arduino UNO comparando con el código realizado de forma tradicional a través del IDE de Arduino.



Desarrollar un DSL que cumpla con las métricas de Usabilidad según el estándar ISO/IEC 9126.

# Objetivos Específicos

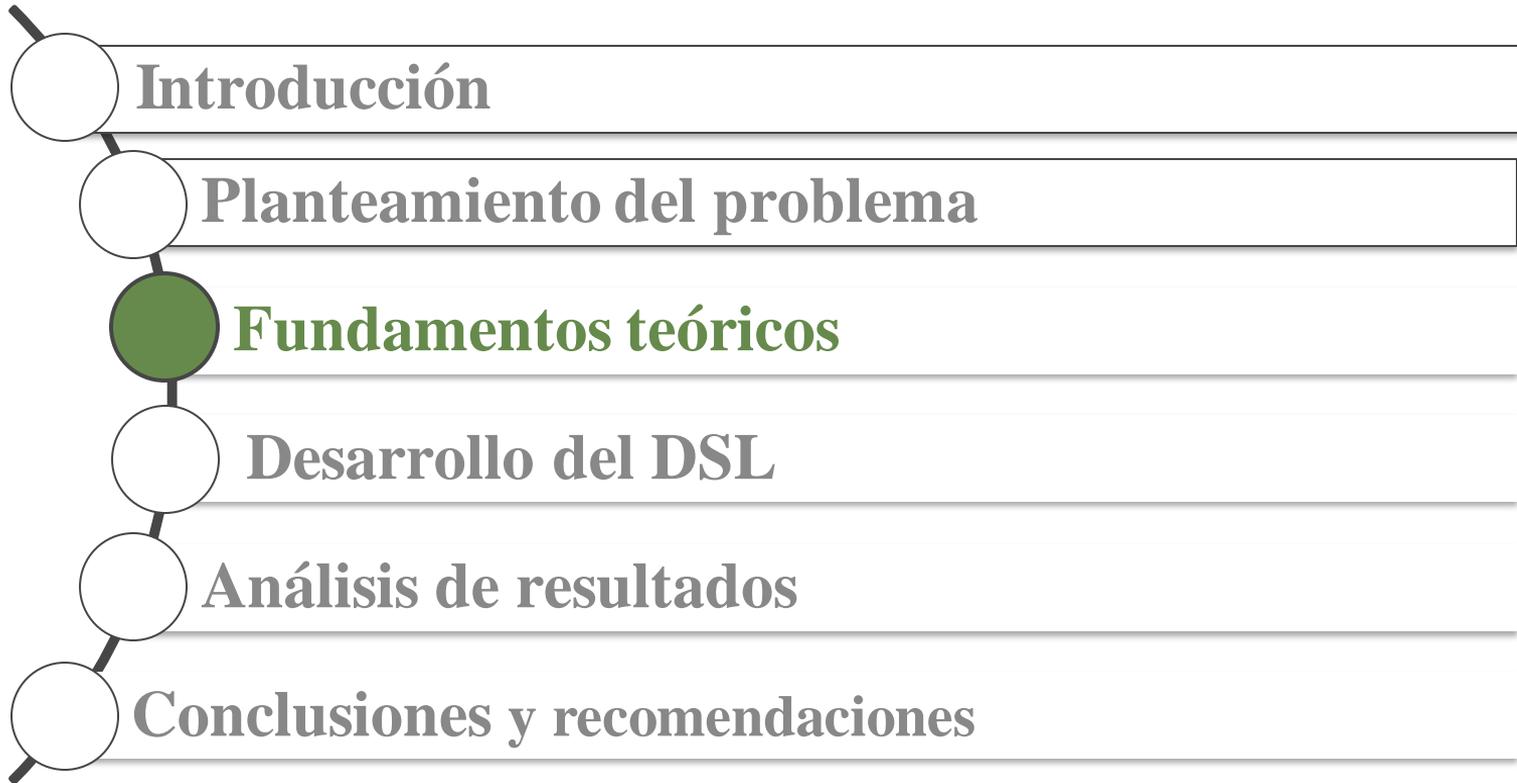


Validar los resultados obtenidos del DSL teniendo un enfoque hacia el cumplimiento de los indicadores señalados, para corroborar que el DSL genera código funcional para programar la placa Arduino UNO utilizando compuertas lógicas “AND”, “OR” y “NOT”.

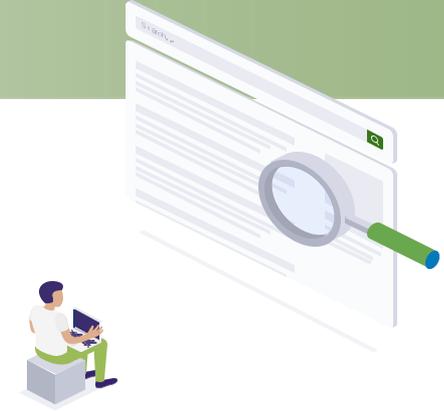
# Hipótesis

Si se desarrolla un DSL se optimizará el tiempo de programación de código funcional para la placa Arduino UNO utilizando compuertas lógicas “AND”, “OR” y “NOT”.





# Entremos en Materia



## Ingeniería Dirigida por Modelos (MDE)

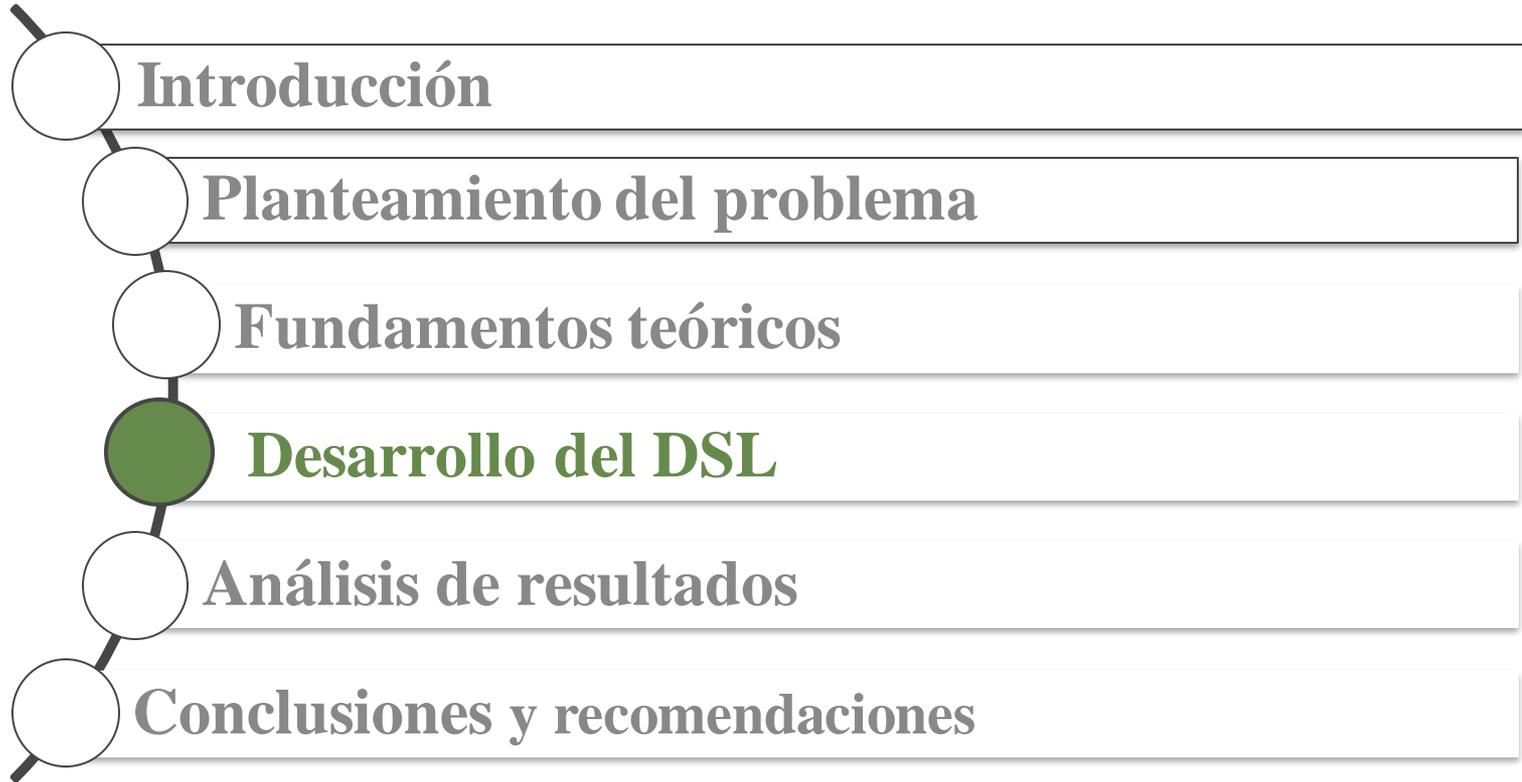
Es una disciplina dentro de la Ingeniería del Software que tiene por objetivo dar soporte a las actividades del ciclo de vida del software, utilizando los modelos como principal artefacto.

## Lenguaje de Dominio Específico (DSL)

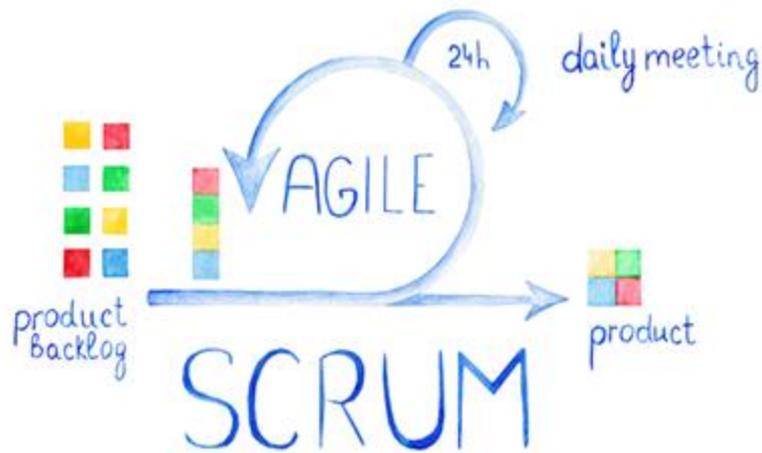
Un DSL es un lenguaje de modelado, gráfico o textual, que se usa para describir una determinada parcela de la realidad.

## Herramientas

- Ecore.
- Sirius.
- Acceleo.



# Metodología



## BACKLOG

en la lista [BackLog](#)

### Descripción [Editar](#)

Tareas a desarrollar  
Duración Sprints -> 4 semanas

### Checklist

0%

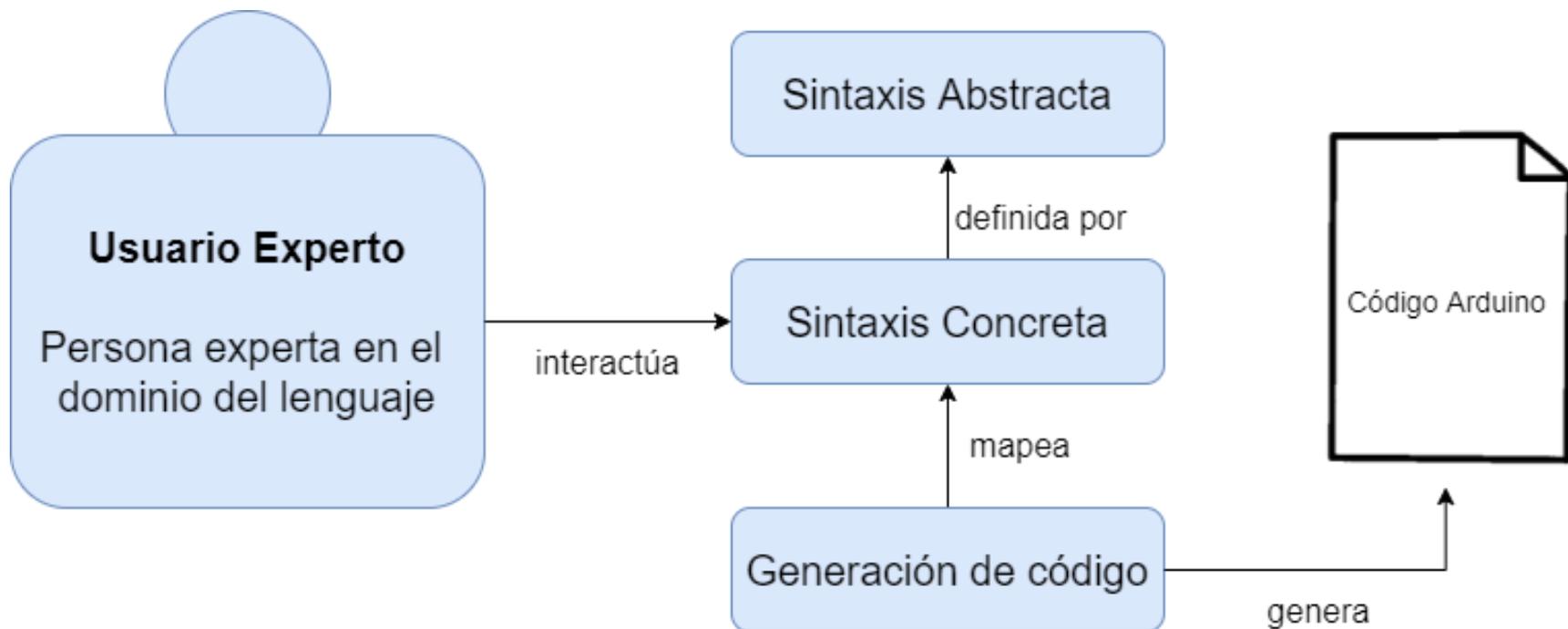
- Creación del metamodelo
- Sintaxis Concreta -> Sirius
- Acceleo -> Generación de código
- Elaboración de pruebas

[Añada un elemento](#)

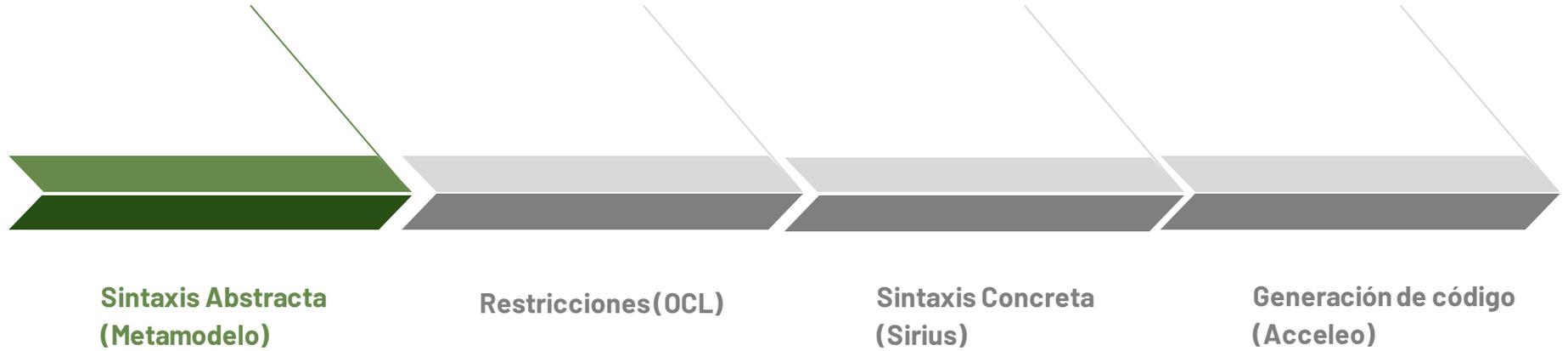


**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

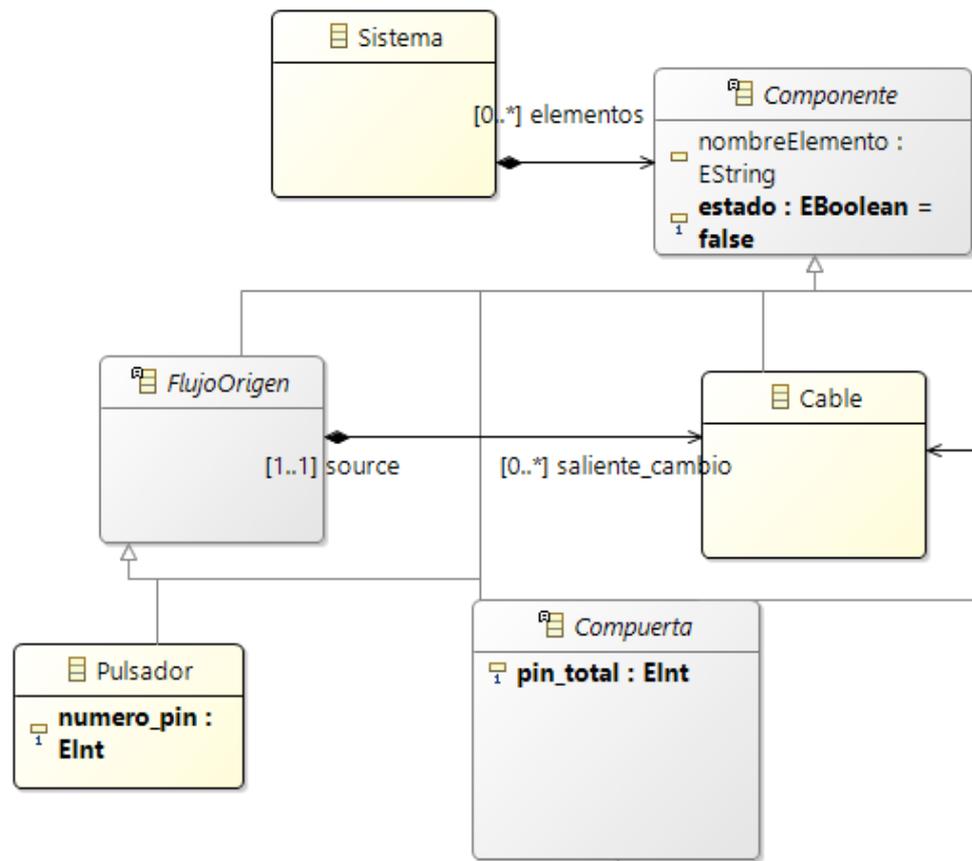
# Diseño Arquitectónico



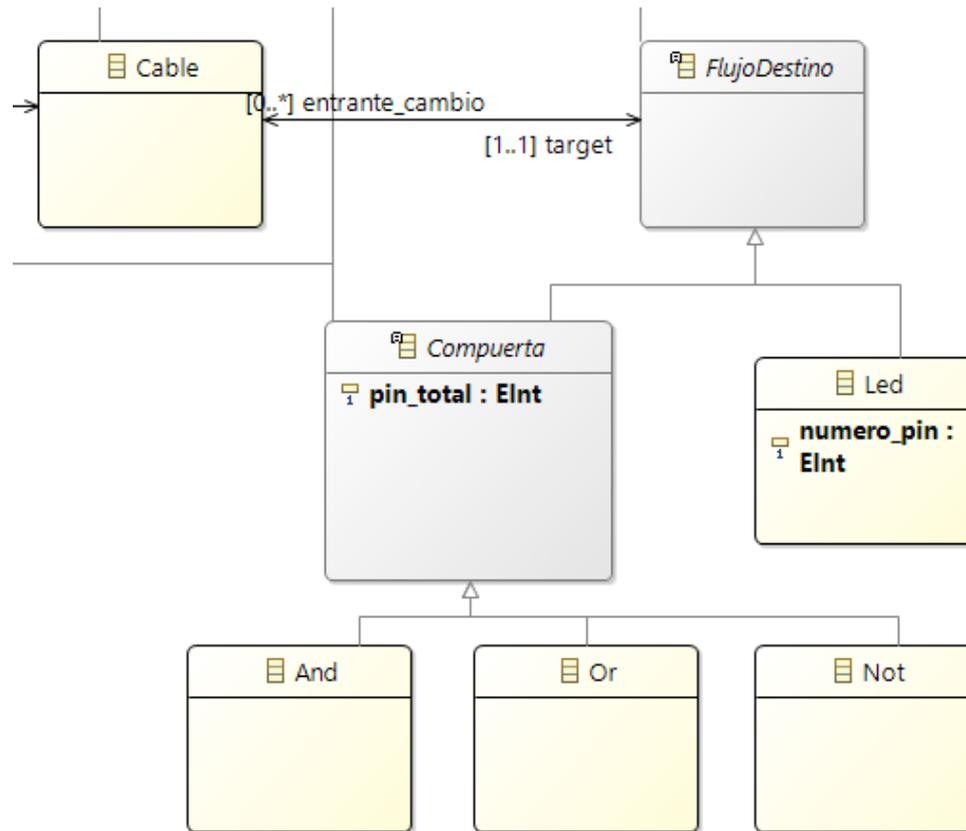
# Desarrollo del DSL



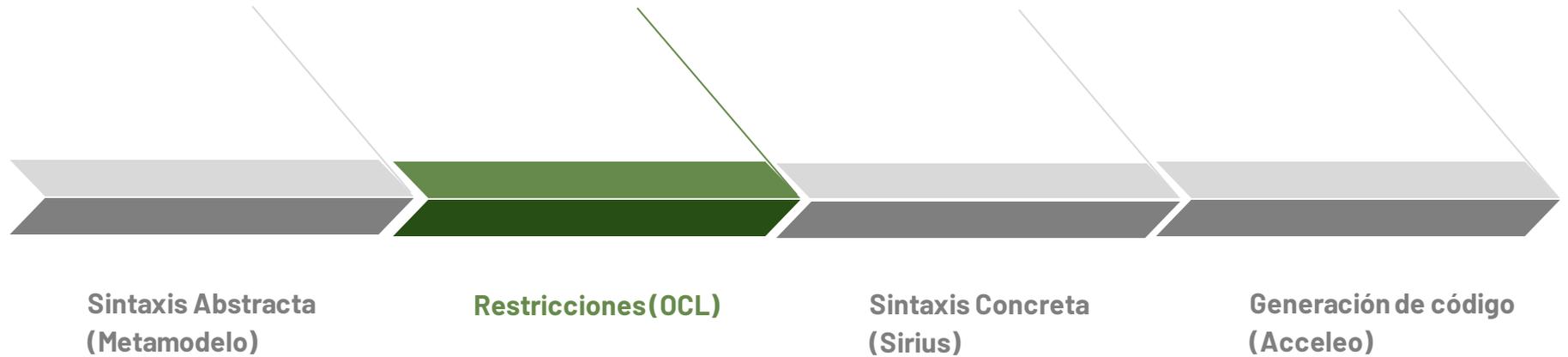
# Sintaxis Abstracta (Metamodelo)



# Sintaxis Abstracta (Metamodelo)



# Desarrollo del DSL

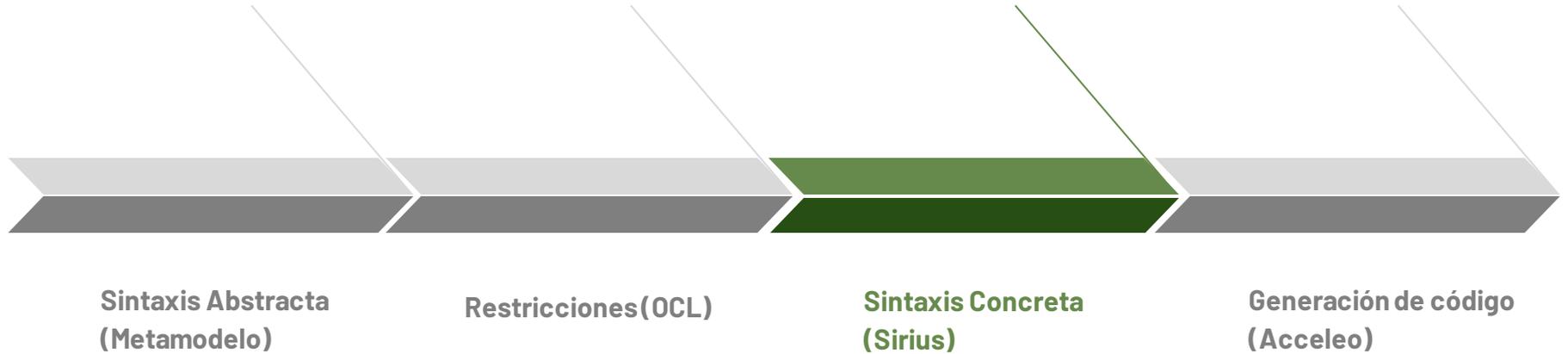


# Restricciones (OCL)

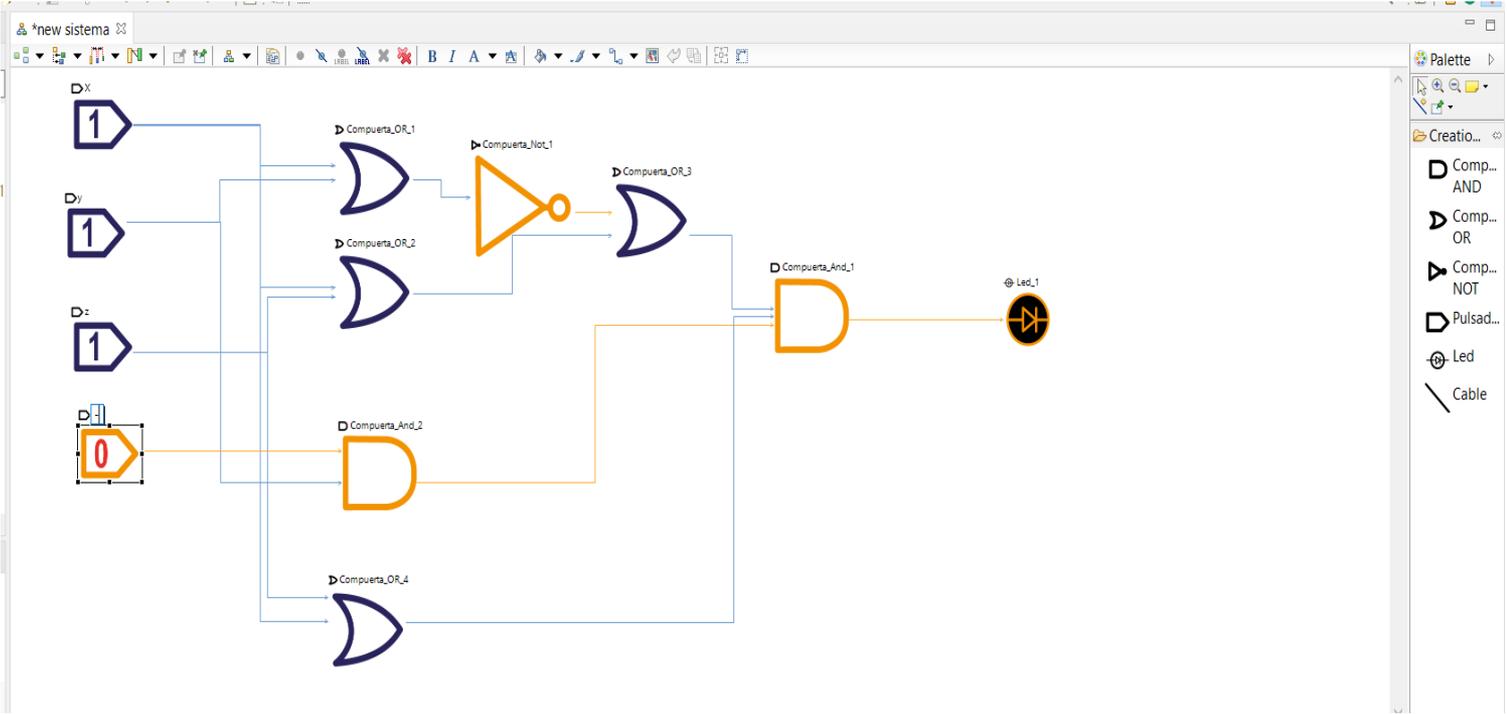
## Validación atributo nombre

```
class Sistema
{
  property elementos : Componente[*|1] { ordered composes };
  invariant
  vacioNombre: self.elementos->forAll(c1 | c1.nombreElemento<>' ' and c1.nombreElemento<>null and c1->excludes(Cable));
  invariant nombre_repetido: self.elementos->forAll(c1,c2 | c1.nombreElemento=c2.nombreElemento implies c1=c2);
}
```

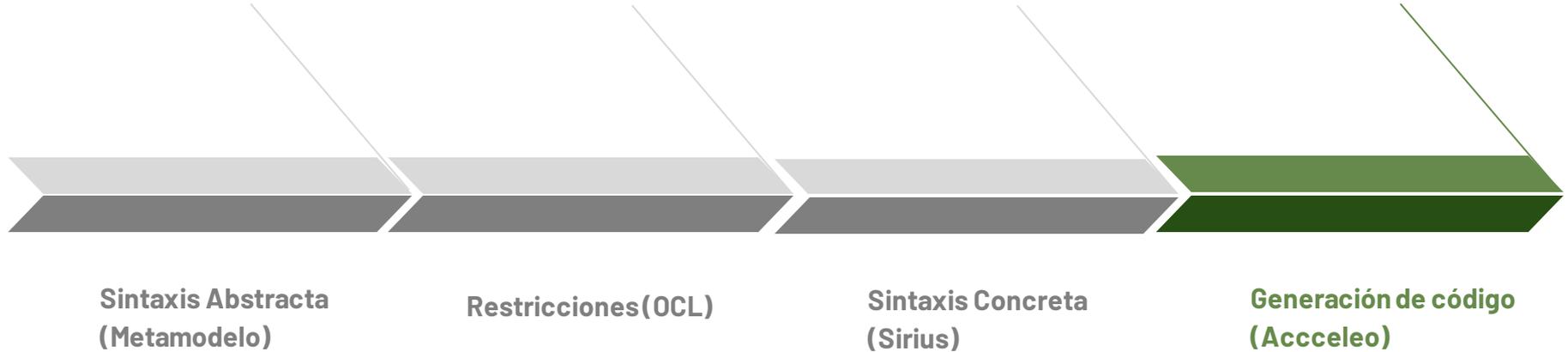
# Desarrollo del DSL



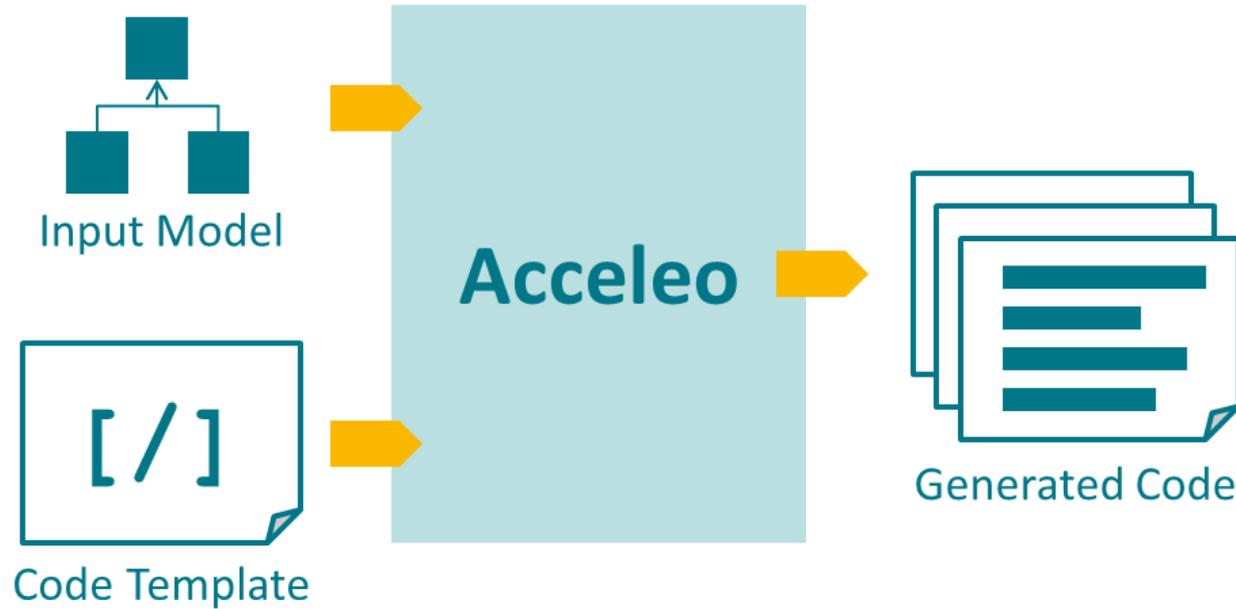
# Sintaxis Concreta (Sirius)



# Desarrollo del DSL

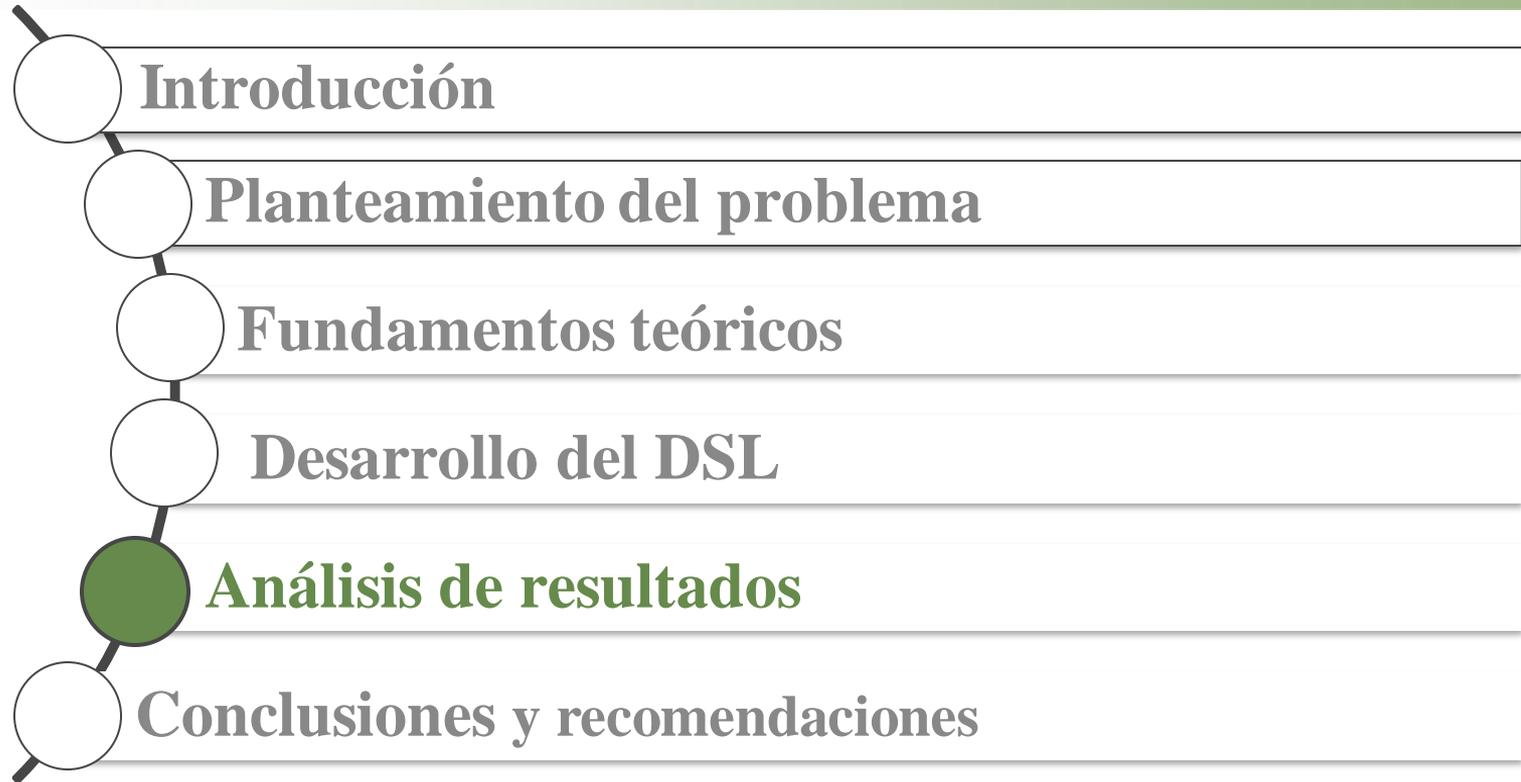


# Generación de código (Acceleo)





[Jrfranco2/circuitoV3 \(github.com\)](https://github.com/Jrfranco2/circuitoV3)



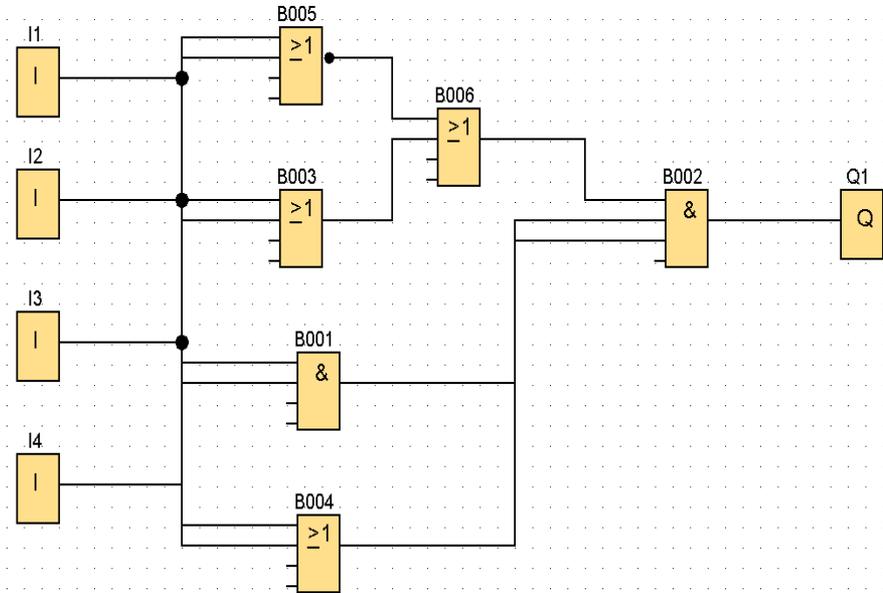
# Evaluación Empírica

- Validar los resultados obtenidos.
- Comparación del DSL desarrollado con: LogoSoft y Visualino con el objetivo de medir la complejidad de las tres soluciones, en términos de tiempo y generación de código funcional.



# Evaluación Empírica

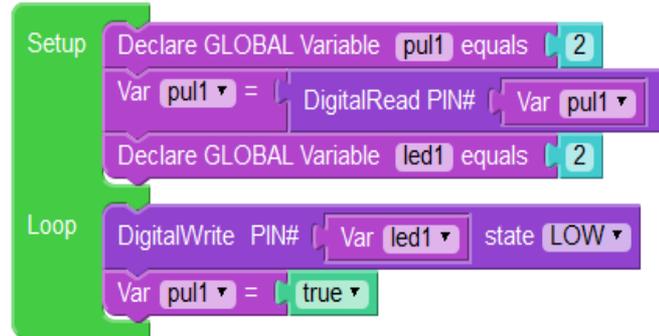
**SIEMENS**  
*Ingenuity for life*



**Visualino**  
powered by RoboBlocks

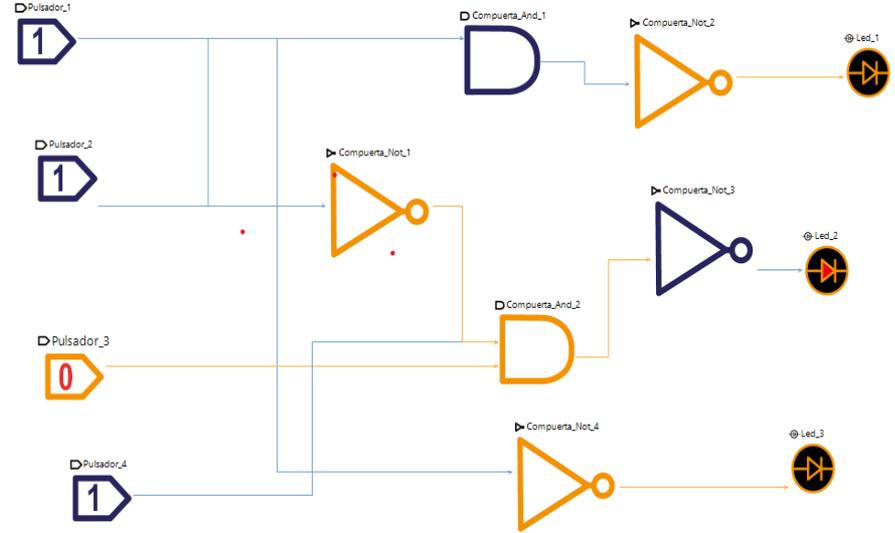
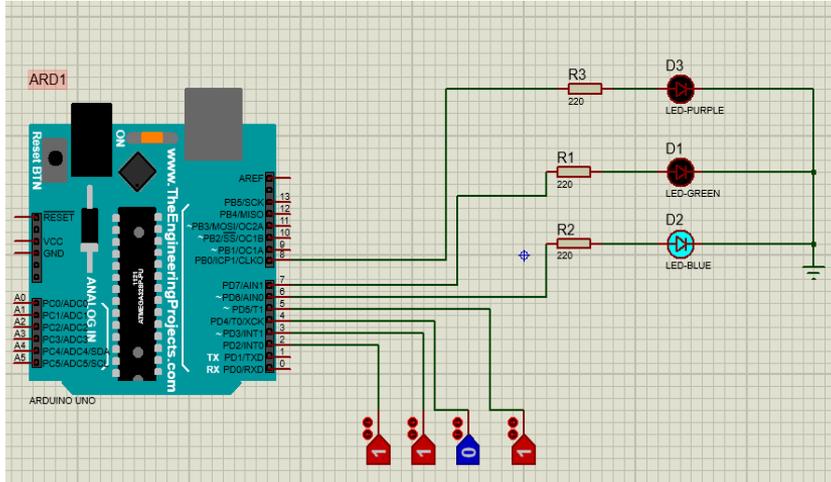
Visual programming environment for Arduino

Open Save Preferences

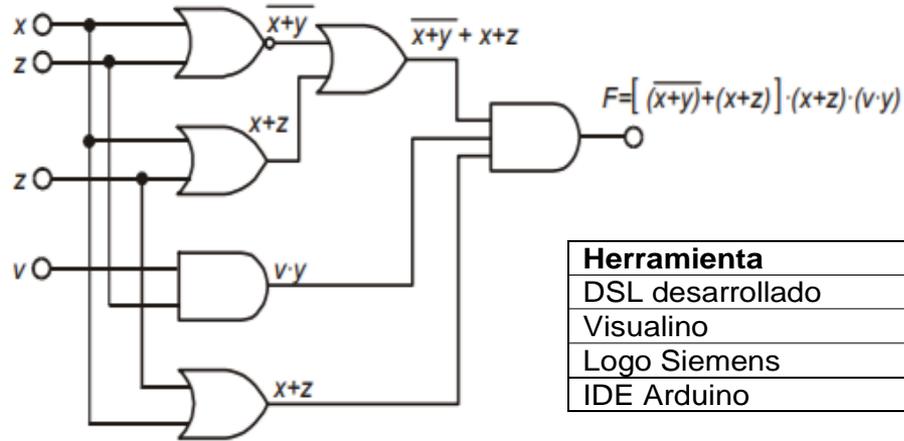


**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

# Evaluación Empírica



# Evaluación Empírica



Herramienta	Líneas de código generadas
DSL desarrollado	109 líneas
Visualino	90 líneas
Logo Siemens	N/A
IDE Arduino	60 líneas

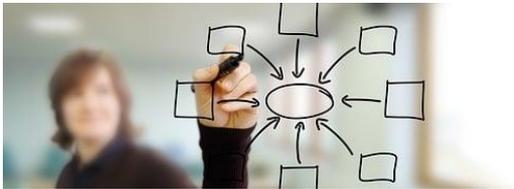
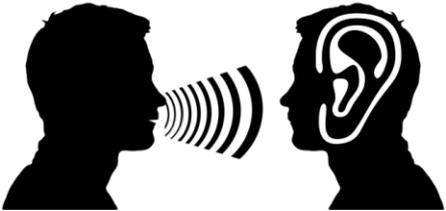
Herramienta	Tiempo empleado
DSL desarrollado	5 min
Visualino	17 min
Logo Soft	3 min
IDE Arduino	8 min

# Evaluación de Usabilidad

- Creación de herramienta gráfica.
- Evaluación según ISO/IEC 9126.
- Conocer el grado de operabilidad.



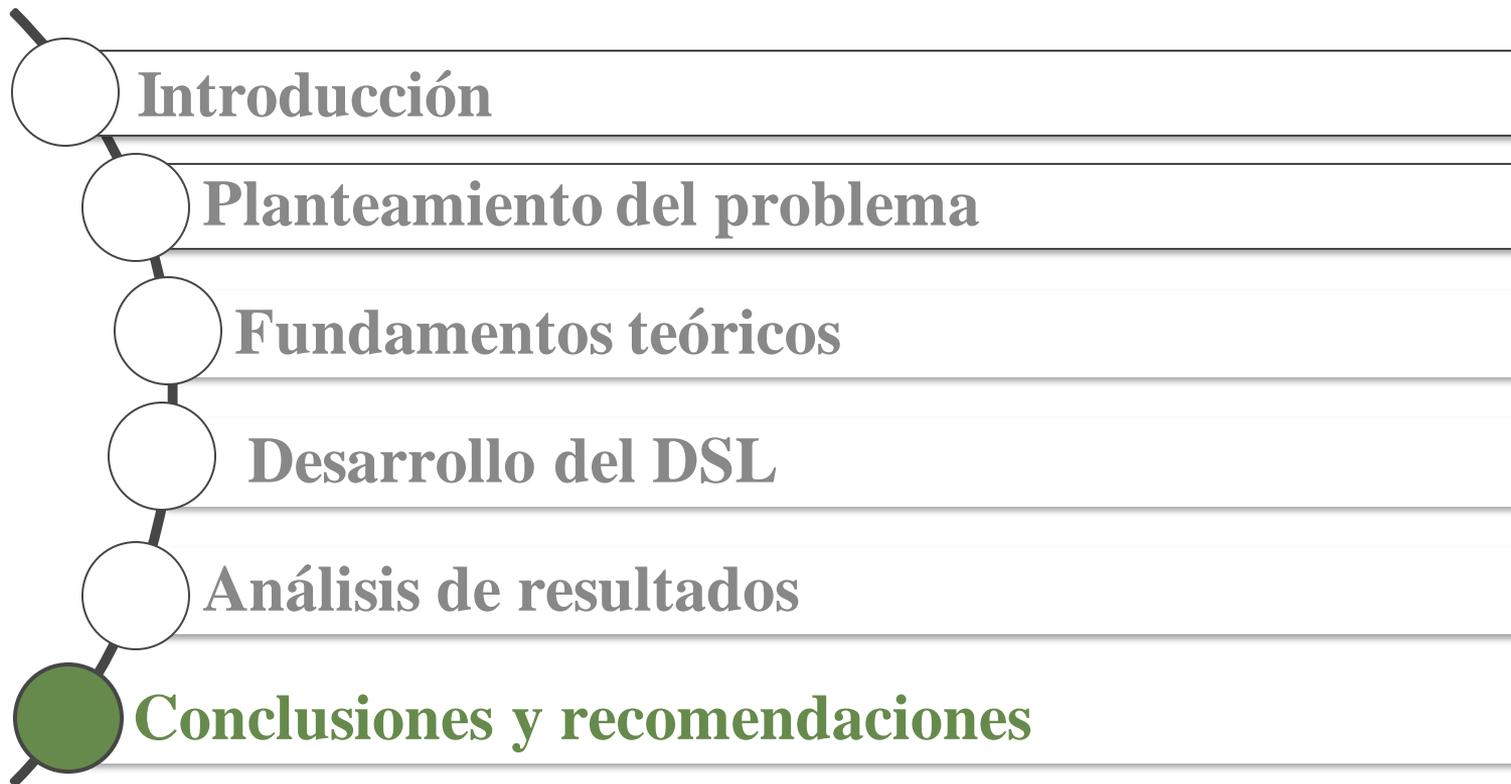
# Evaluación de Usabilidad



# Validación de la hipótesis

Se desarrolló un DSL que permite diseñar circuitos electrónicos en base a compuertas lógicas AND, OR y NOT; lo cual ha demostrado que se puede optimizar el tiempo de programación de código funcional para la placa Arduino UNO.





# Conclusiones

- Se desarrolló un DSL que permite diseñar circuitos electrónicos en base a compuertas lógicas AND, OR y NOT; lo cual ha demostrado que se puede optimizar el tiempo de programación de código funcional para la placa Arduino UNO.
- El DSL es una herramienta al usuario final (experto de dominio) que permite desarrollar software sin que este sea experto en los típicos lenguajes de programación porque le permite manejar conceptos de su cotidianidad y no sintaxis de un lenguaje de programación.

# Conclusiones

- El editor gráfico utiliza nomenclatura electrónica propia de compuertas lógicas, pulsadores y leds.
- El generador de código desarrollado toma en cuenta las posibles combinaciones de los elementos y su interacción.
- Por la manera que se encuentra construido el metamodelo la escalabilidad del aplicativo es posible gracias a los flujos de salida y entrada que se maneja entre las diferentes clases.

# Recomendaciones

- Para desarrollar software se recomienda la utilización de un DSL, porque permite desarrollar software utilizando un nivel más alto de abstracción, debido a que permite utilizar términos y definiciones más apegadas al problema que se quiere resolver, utilizando la notación que maneja el experto del dominio (usuario del software)
- Implementar una interfaz gráfica amigable al usuario, en la cual se respeten los símbolos propios del área del problema, porque de esta manera se reduce mucho el tiempo de aprendizaje del DSL.

# Recomendaciones

- Generar la plantilla personalizada de Acceleo por segmentos, puesto que en la etapa de generación de código se trabaja con el modelo desarrollado y de esta manera no se mezclan elementos de otra sección.



Gracias por su atención

# Preguntas....

