



XVI Congreso de **Confiability**

San Sebastián, 3 y 4 de diciembre de 2014

Aumento de Confiability de un Sistema de Control de Cogeneración Basado en Mejoras de TEICs.



Indice



1. Introducción – Contexto del Trabajo
2. Objetivos del Trabajo
3. Desarrollo del Trabajo
4. Testeos Realizados y Resultados Obtenidos
5. Conclusiones





Introducción- Contexto del Trabajo



- Mondragon Goi Eskola Politeknikoa

Escuela Politécnica Superior

MU Estudios Investigación Servicios Sobre nosotros Actualidad MU Komunitatea

DIMENSIÓN CIENTÍFICO TECNOLÓGICA

- Comportamiento mecánico y diseño de producto
- Ciencia, Tecnología y Procesos de transformación de materiales
- Organización y gestión industrial
- Sistemas embebidos
- Sistemas de Información
- Energía Eléctrica

DIMENSIÓN SECTORIAL / EMPRESARIAL

- Máquina-Hita
- Automoción
- Aeronáutica
- Ferroviano
- Elevación
- Energía
- Electromésicos
- TICs
- Salud
- Construcción

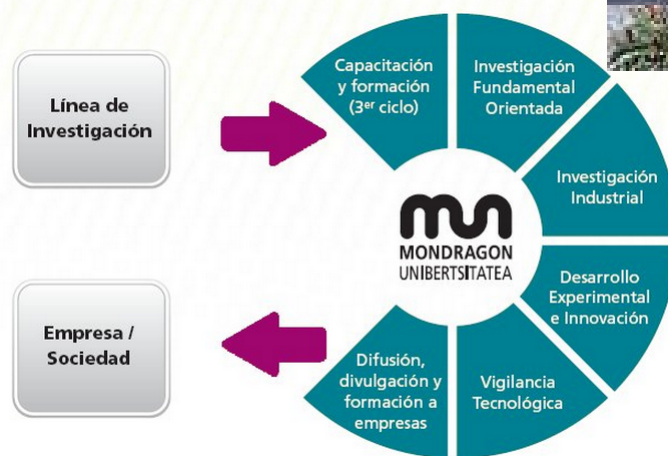
Programas I+D empresarial y sectorial

Lineas de investigación

- Línea de Investigación
- Empresa / Sociedad



www.mondragon.edu/eps



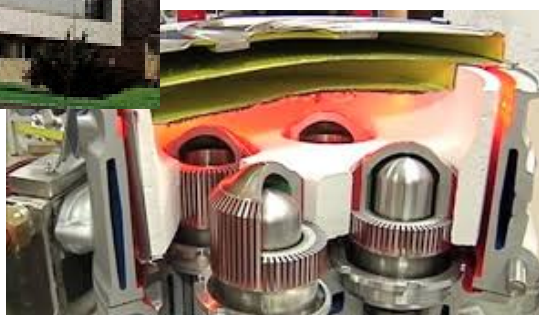


Introducción- Contexto del Trabajo



- Centro Stirling

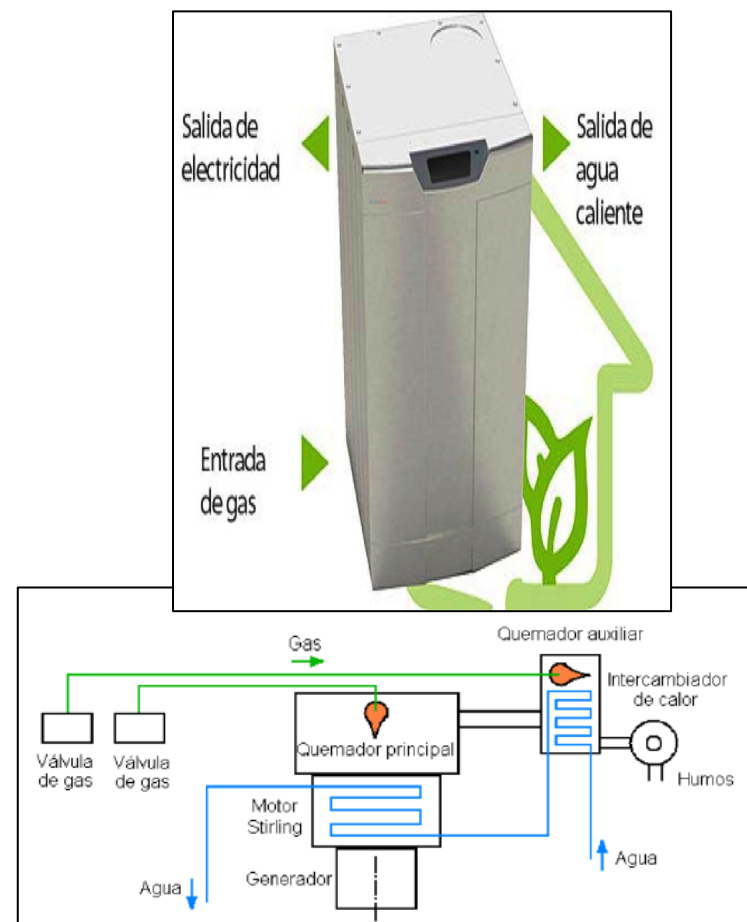
Centro tecnológico dedicado a las energías sostenibles en el entorno del hogar de Mondragón Componentes, perteneciente a la Corporación Mondragón.





Contexto de Trabajo

- Tecnología heredada
- Funcionamiento:
 - Generador que produce electricidad a 230 V de corriente alterna.
 - Motor Stirling que crea la energía motriz del generador (Core Engine).
 - Quemador de gas acoplado que produce el calor que mueve el motor Stirling (Main Burner).
 - Quemador auxiliar que produce calor adicional durante los picos de demanda (Boost Burner).
 - Intercambiador de calor que recupera el calor de los gases de combustión producidos en el quemador (Heat Exchanger).
 - Dos ventiladores que suministran la mezcla de gases a los quemadores, y al mismo tiempo facilitan la salida de los humos de combustión.
- **Necesidad de entender producto (SW) y uso de Model Based Development**





Objetivos del Trabajo

- Desarrollar el control del software del sistema WhisperGen en la herramienta Enterprise Architect y poder generar código a partir del comportamiento de los diagramas UML.
- Pasar el software generado por Enterprise Architect al hardware y testearlo.
- Reorganizar el SW del sistema de control teniendo en cuenta la reusabilidad para poder utilizar en otros productos.



Desarrollo del Trabajo



- Modelado del Sistema en Enterprise Architecture
 - Diagrama de Clases
 - Diagrama de Estados → Comportamiento del Sistema
 - Reorganización del Sistema
 - Cada componente → Funcionalidad concreta
- **Favorecer la Reusabilidad**





Desarrollo del Trabajo



- Generación de Código Automático
 - Problemas: EA v10
 - No soporta regiones.
 - No código para soporte de eventos/timers
 - Análisis diferentes alternativas:
 - SinelaboreRT
 - AMUSE: plugin de EA
 - Enar Uml2Code
 - EA v11





Desarrollo del Trabajo

- Comparación

SINELABORE RT	ENAR UML2CODE	EA VERSIÓN 11
<ul style="list-style-type: none">+ Código simple y manejable+ Soporta estados jerárquicos, regiones, eventos	<ul style="list-style-type: none">+ Código simple+ Soporta estados jerárquicos, regiones, eventos	<ul style="list-style-type: none">+ Soporta estados jerárquicos, regiones, eventos+ Misma herramienta para la generación del código
<ul style="list-style-type: none">- No genera código del timer- Limitado a 4 estado jerárquicos- Diferente herramienta- Gastos: Compra de la licencia	<ul style="list-style-type: none">- No genera código del timer- Diferente herramienta-Gastos: compra del SW	<ul style="list-style-type: none">-Código complejo y difícil de entender-No genera código del timer



Desarrollo del Trabajo

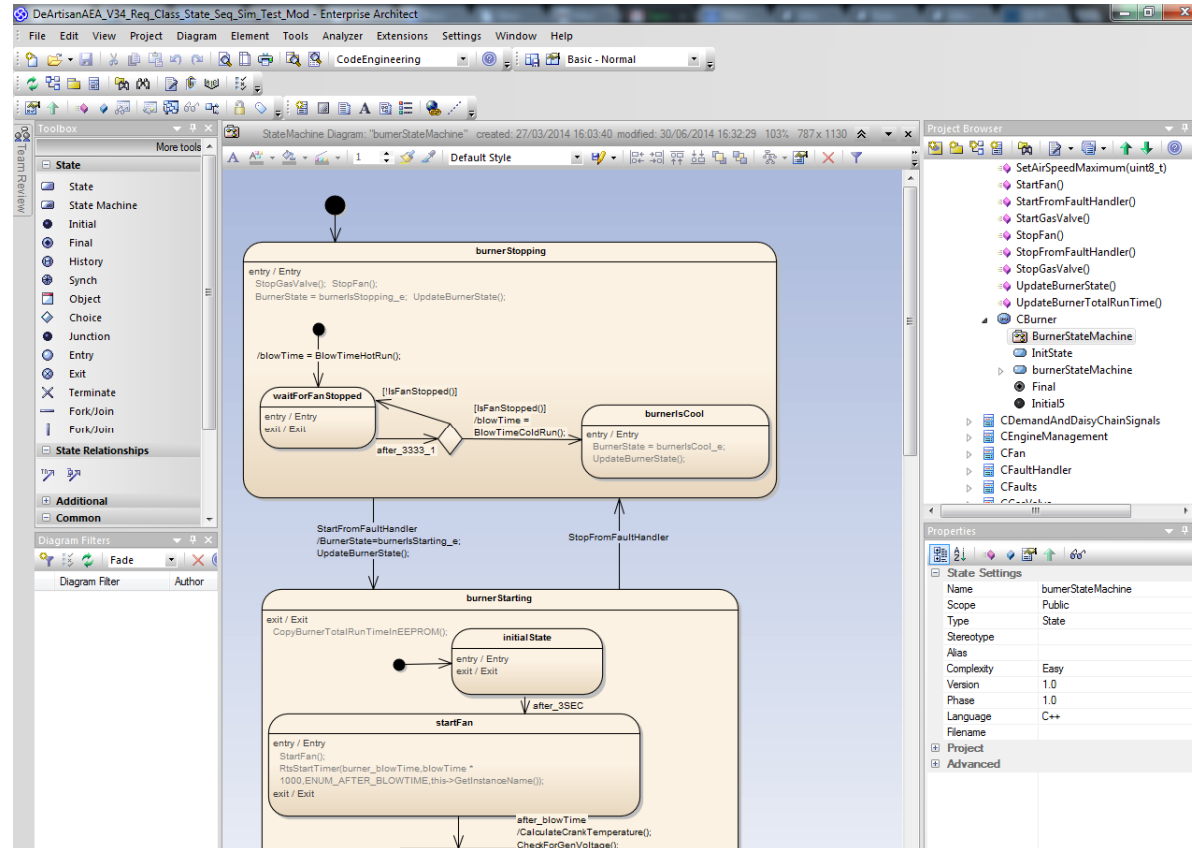
• Comparación

SINELABORE RT	ENAR UML2CODE	EA VERSIÓN 11
<ul style="list-style-type: none">+ Código simple y manejable+ Soporta estados jerárquicos, regiones, eventos	<ul style="list-style-type: none">+ Código simple+ Soporta estados jerárquicos, regiones, eventos	<ul style="list-style-type: none">+ Soporta estados jerárquicos, regiones, eventos+ Misma herramienta para la generación del código
<ul style="list-style-type: none">- No genera código del timer- Limitado a 4 estado jerárquicos- Diferente herramienta- Gastos: Compra de la licencia	<ul style="list-style-type: none">- No genera código del timer- Diferente herramienta-Gastos: compra del SW	<ul style="list-style-type: none">-Código complejo y difícil de entender-No genera código del timer



Desarrollo del Trabajo

- EA v11





Desarrollo del Trabajo



- **Generación Código EA v11**
 - **Diseño de diagramas UML del Control del Sistema Whispergen**
 - **Crear Máquina de Estados Ejecutable**
 - Una instancia para cada clase del proyecto.
 - **Clases generadas:**
 - **Clases del proyecto**
 - **Clases motor EA** (ContextManager, EventProxy, SimulationManager, Console)





Testeos y Resultados Obtenidos



- Testeos a la generación automática
 - Estados Jerárquicos
 - Regiones
 - Eventos
 - Timers
 - Estados de History





Testeos y Resultados Obtenidos

- Test Recursos de Memoria
 - ATmega2560

```
177 817 bytes of CODE memory  
18 361 bytes of DATA memory (+ 138 absolute )  
298 bytes of XDATA memory
```

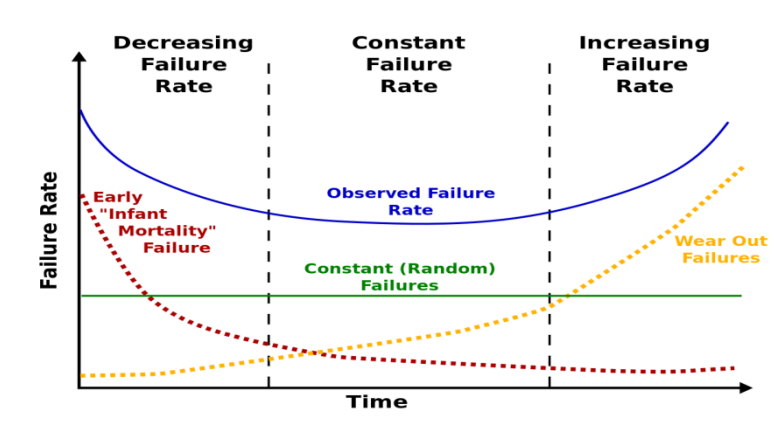
32 Registers	\$0000 - \$001F
64 I/O Registers	\$0020 - \$005F
416 Ext I/O Reg.	\$0060 - \$01FF
Internal SRAM (8192 x 8)	\$0200
External SRAM (0 - 64K x 8)	\$21FF \$2200

- .hex → 489Kb. Dispone de recursos necesarios



Conclusiones

- Adquisición del conocimiento del sistema
- Generación Automática del Código del SW de control (EA v11)
- SW del sistema organizado en módulos verificados
 - Aumenta Reusabilidad
 - Aumenta Confiabilidad





Eskerrik Asko!
Gracias!
Thank you!

