

PROMETEO

Plataforma Tecnológica Española de
Sistemas con Inteligencia Integrada



Entorno para operación y validación de sistemas embebidos

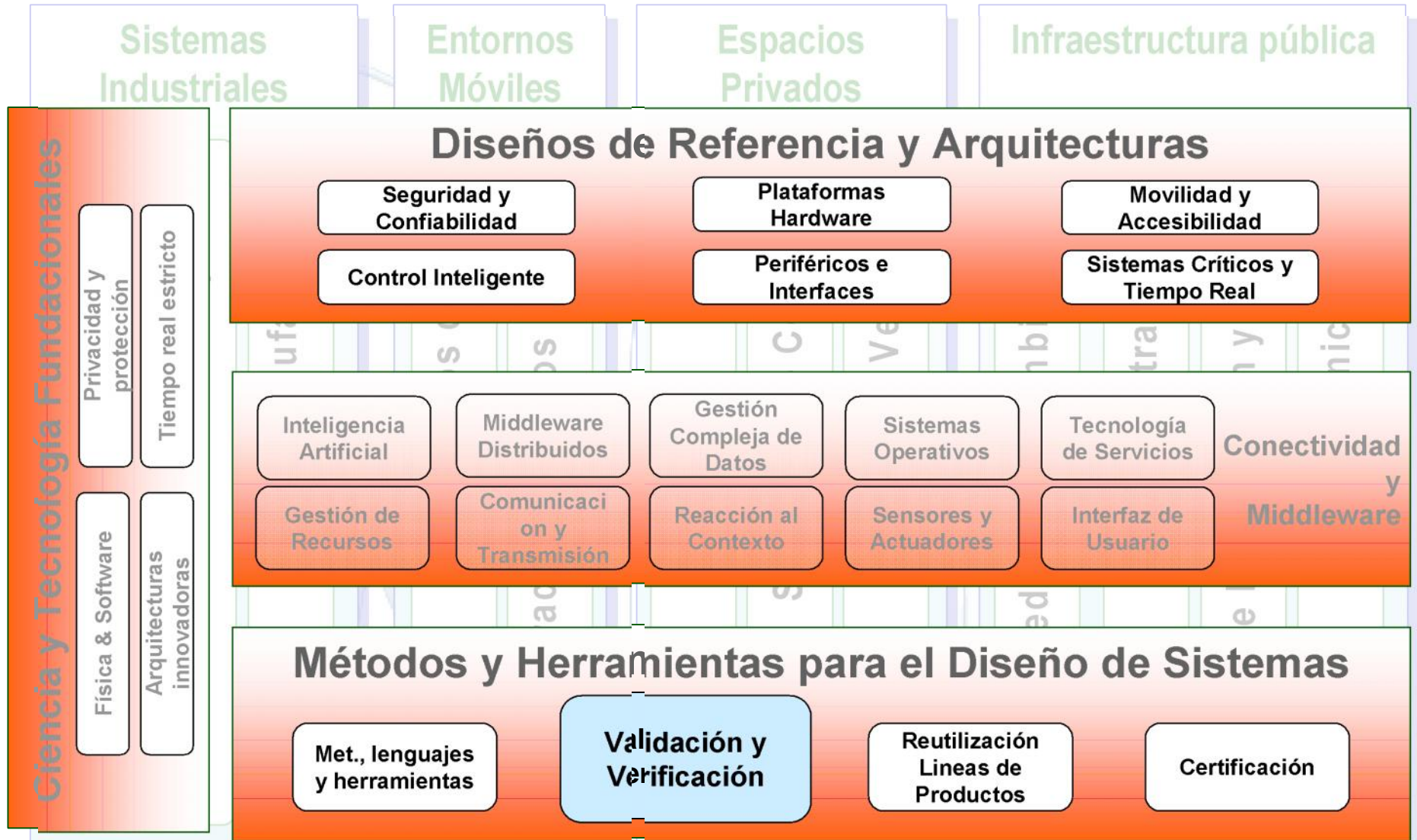
Agustín Yagüe Panadero (ayague@eui.upm.es)

Talleres:

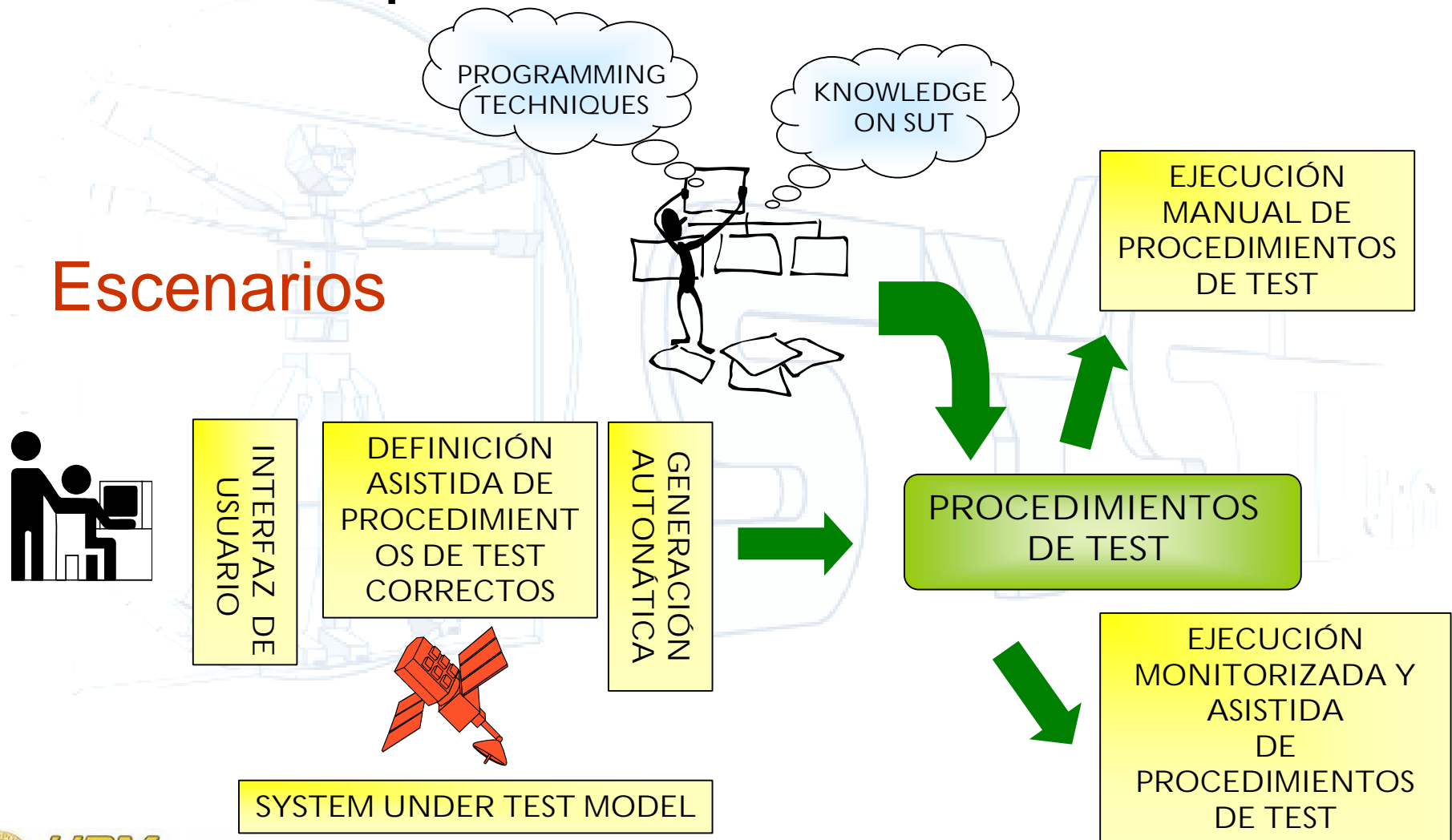
- Diseños de Referencia y Arquitecturas
- Métodos y Herramientas para el Diseño de Sistemas

San Sebastián
18 diciembre 2006

Contexto de Prometeo



Entorno para pruebas de validación y operación de sistemas



Entorno para pruebas de validación y operación de sistemas

- **Actividad de testing de sistemas**

- Compleja y sofisticada
- Con software embebido
- Y requiriendo cero defectos

Es necesario probar el comportamiento dinámico del sistema completo, no es suficiente probar cada elemento independientemente

- **Objetivo**

- Herramientas que puedan ser adaptadas a diferentes sistemas industriales

Es esencial definir una arquitectura independiente del dominio de aplicación

Entorno TOPEN existente

- Entorno basado en el modelo del sistema a validar (SUT: System Under Test)
- Ayuda a definir los procedimientos de prueba de una manera independiente del lenguaje
- Gestiona su ejecución en el *SUT* de manera **distribuida y remota**
- Almacena los resultados para una posterior análisis o monitorización

Características de TOPEN

- Simplifica el proceso de definición y ejecución de pruebas en sistemas complejos
- Diseñado para validar el comportamiento de componentes con software embebido
- Entorno distribuido que permite tele-testing de sistemas con elementos interconectados
- Permite definir pruebas a nivel de usuario
- Adaptación factible a múltiples sistemas industriales

Ejemplo de entorno

The screenshot displays a software interface for a test environment, titled "TEST OPERATION ENVIRONMENT". It features a menu bar with "FILE", "SESSION", "WORKING MODE", "HELP", "LOCATION", and "VIEW".

The interface is divided into several windows:

- MANAGER PANEL INTERFACE:** Shows a logical view of the system under test, represented by a tree structure with three nodes. A yellow box labeled "System Under Test (logical view)" points to this window.
- Test Procedure Panel Interface - TESTPROCEDURE11:** Contains a list of test steps:

```
2 SET 3 1 CreditCounter 50;
2 SET 2 CreditCounter 40;
2 START 2 3 1;
2 Wait 0 Jackpot yes;
2 GET 1 CreditCounter;
2 GET 2 CreditCounter;
2 GET 3 CreditCounter;
2 SET 3 1 CreditCounter 500;
2 SET 2 CreditCounter 400;
```

A yellow box labeled "Test Procedure" points to this window.
- Execute Result Interface:** Displays the results of the test execution:

```
- Result: SET 1 CreditCounter 50 Ok
- Result: SET 2 CreditCounter 40 Ok
- Result: START 2 Ok
- Result: START 3 Ok
- Result: START 1 Ok
- Notification: Neld= 1 LOCKED = ON
- Notification: Neld= 2 LOCKED = ON
- Notification: Neld= 3 LOCKED = ON
- Notification: Neld= 2 JACKPOT = 123.0
- Result: WAIT 0 Jackpot yes Ok
- Result: GET 1 CreditCounter 34
- Result: GET 2 CreditCounter 28
- Result: GET 3 CreditCounter 38
- Result: SET 3 CreditCounter 500 Ok
- Result: SET 1 CreditCounter 500 Ok
- Result: SET 2 CreditCounter 400 Ok
- Notification: Neld= 0 DISCONNECT = yes
```

A yellow box labeled "Execution Results" points to this window. A blue box labeled "Response" points to the first three lines of results, and another blue box labeled "Notification" points to the subsequent lines.

Ejemplo de Gramática

```
TP Tpld;
ManagerId set Neld + { creditCounter Value };
ManagerId getNeld + { creditCounter | jackPot };
ManagerId start Neld + ;
ManagerId finish Neld + ;
ManagerId wait Condition;
for ManagerId Number times Statements endFor ;
repeat {until | while} ManagerId Condition Statements endRepeat ;
ManagerId createNe Neld { classId {Master | Accumulator }
    physicalId PhysicalNeld};
ManagerId deleteNe Neld;
EndTP;
```

Ejemplo de Procedimiento de Prueba

Requisitos de sistema

Cuando una máquina recibe un comando start, tienen que comenzar las jugadas

• Especificación de la Prueba

- **Datos de entrada**
1500 creditos en máquina 1
- **Proceso**
Arrancar máquina 1
- **Datos de salida**
Contador de créditos en la máquina 1
- **Resultado esperado**
Contador de crédito inferior a 1500

TOPEN testing and operation

Procedimiento de Test

```
2 SET 1 CreditCounter 1500;  
2 START 1;  
2 FINISH 1;  
2 GET 1 CreditCounter;
```

Resultados

pass

- Result: SET 1 CreditCounter 1500 Ok
- Result: START 1 Ok
- Result: FINISH 1 Ok
- Result: GET 1 CreditCounter 1499

Procedimientos de Test

Requisito

Un premio jackpot tiene que ocurrir antes de 1000 jugadas en una máquina
(un crédito – una jugada)

Procedimiento de Test

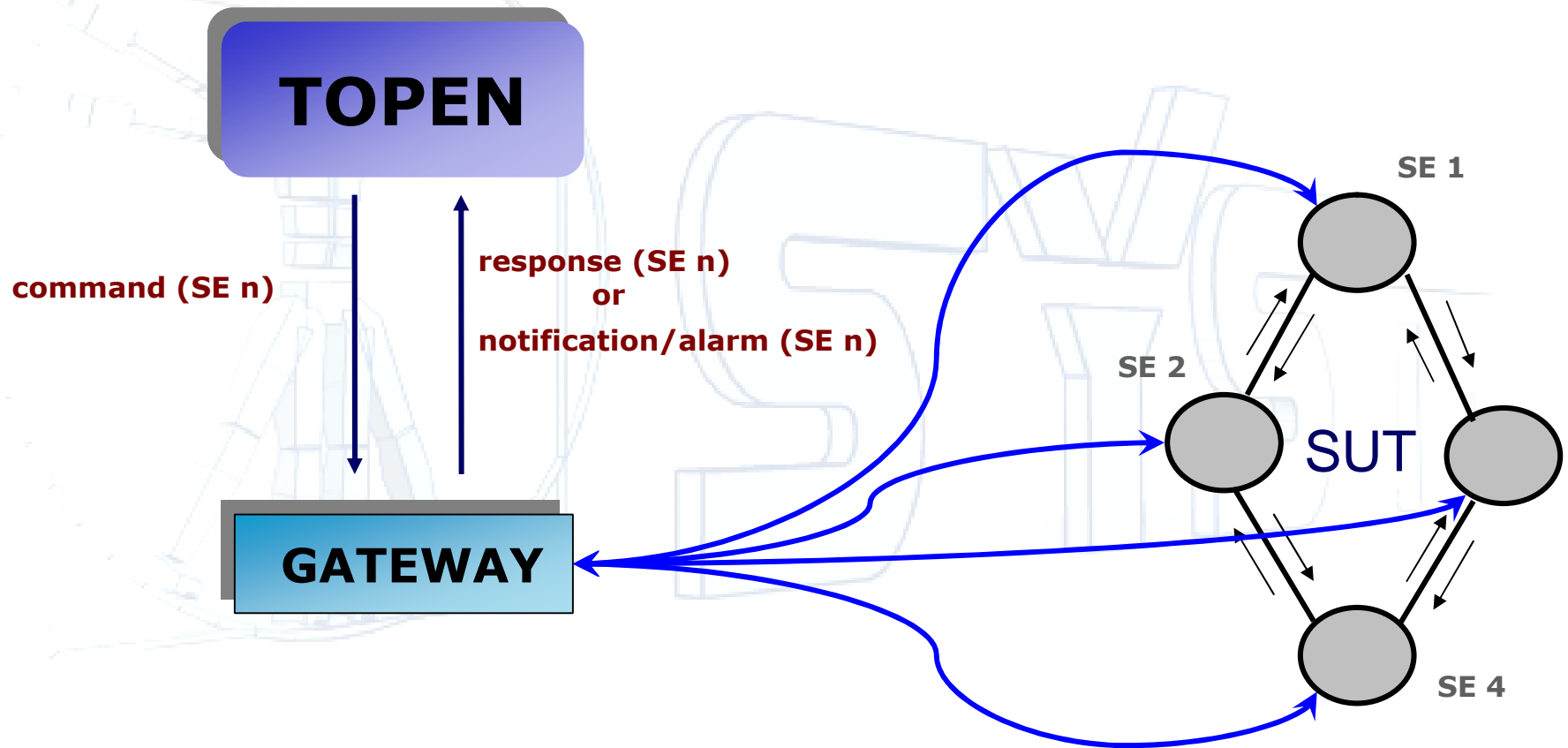
```
2 SET 1 2 3 CreditCounter 1000;  
2 START 1 2 3;  
2 WAIT 0 JackPot yes;  
2 GET 1 2 3 CreditCounter;
```

1
2
n

```
Result: SET 1 CreditCounter 1000 Ok  
Result: SET 1 CreditCounter 1000 Ok  
Result: SET 1 CreditCounter 1000 Ok  
Result: SET 2 CreditCounter 1000 Ok  
Result: SET 3 CreditCounter 1000 Ok  
Result: START 1 Ok  
Result: START 2 Ok  
Result: START 3 Ok  
Notification: NeId= 1 LOCKED = ON  
Notification: NeId= 2 LOCKED = ON  
Notification: NeId= 3 LOCKED = ON  
Notification: NeId= 3 JACKPOT = 202.0  
Result: WAIT 0 JackPot yes Ok  
Result: GET 1 CreditCounter 701  
Result: GET 2 CreditCounter 705  
Result: GET 3 CreditCounter 705
```

Un procedimiento de test debería ejecutarse n veces

Arquitectura TOPEN



Arquitectura TOPEN

- Componentes de TOPEN
 - Man Machine Interface
 - Interface Control Manager (ICM)
 - Manager Panel Interface (MPI)
 - Test Procedure Panel Interface (TPPI)
 - Mission Information Base
 - MIB component
 - MIB data base (relational data base)
 - Code Generator (Topen Engine)
 - Code Generator
 - Code Generator Communicator
 - Agent (Gateway)
 - Agent Communicator
 - Agent (any programming language: C, C++, Java, etc.)
 - System Under Test

Evaluación de resultados de test

Test Analysis Results Application

File View Help

Saved Test Procedure View

Saved Test Procedures:

- TP1_
- TP2_

```

1 SET 1 CREDITCOUNTER 1000 ;
1 SET 2 CREDITCOUNTER 2000 ;
1 SET 3 CREDITCOUNTER 3000 ;
1 START 1 2 3;
1 WAIT 0 JACKPOT YES;
    
```

Type Evaluation

Type Value

EXPECTED RESULTS

SE	Attribute	Value	Min Value	Max Value
1	JACKPOT	TRUE		
2	JACKPOT	TRUE		
3	JACKPOT	TRUE		

Add Expected Result Delete Expected Result Update Expected Results Set Check Automatically

EXECUTIONS (select an execution, please)

Test Procedures	Name	Execution Time	Validated desired results
17	TP1_284058107	25/11/2005-13:57:19	OK
18	TP1_725343024	13/12/2005-11:09:03	OK

Update Validated Desired Results

EXECUTED COMMANDS AND RESULTS

Manager	SE	Command Time	Command	Attribute Name	Attribute Value	Result Time	Result
1	3	25/11/2005-13:...	SET	CREDITCOUNTER	3000	25/11/2005-...	
1	0	25/11/2005-13:...	WAIT	JACKPOT	YES	25/11/2005-...	
1	1	25/11/2005-13:...	GET	JACKPOT		25/11/2005-...	FALSE
1	2	25/11/2005-13:...	GET	JACKPOT		25/11/2005-...	FALSE
1	3	25/11/2005-13:...	SET	JACKPOT		25/11/2005-...	TRUE

NOTIFICATIONS

SE	Notification Time	Name
1	25/11/2005-13:57:30	Locked
2	25/11/2005-13:57:31	Locked
3	25/11/2005-13:57:32	Locked
3	25/11/2005-13:57:33	JackPot

Evolución de TOPEN

El entorno de pruebas y operación TOPEN es de sumo interés para las empresas de cualquier sector industrial que produzca y/o utilice sistemas complejos con software, como por ejemplo:

- Equipos electrónicos con software embebido
- Equipos mecatrónicos
- Redes de ordenadores

Ejemplos de aplicación:

- Energía (Proyecto en realización [OpEnFuel](#))
- Automoción
- Aeronáutica
- Telefonía y comunicaciones
- Industria Aeroespacial
- Domótica
- etc.

Agradecimientos

- Idea previa de U.P.M. (Proyecto ARCO)
- Contacto con Ligeron S.A. (Francia) y Labein, para suministrar casos prácticos y así enriquecer la casuística y mejorar TOPEN
- Resultado: Proyecto IST, de acrónimo **METSES**, financiado parcialmente por la C.E.

Personas de Contacto

- Juan Garbajosa Sopeña
 - mail: jgs@eui.upm.es
 - teléfono: 91 336 50 81
- Agustín Yagüe Panadero
 - mail: jgs@eui.upm.es
 - teléfono: 91 336 50 84
- Pedro Pablo Alarcón Caverro
 - mail: jgs@eui.upm.es
 - teléfono: 91 336 50 83

Página web: <https://syst.eui.upm.es>