TEST AUTOMATIZADOS DEL SISTEMA “GÜERTITO”

Basado en Simulink® Requirements™ Y Simulink® Test™

*Versión 1.0*

***GRUPO F:***

JORGE DEIRO FERRE (M19056)

JOSE FIGUEROLA PALACIOS (M19080)

JUAN RUBIO ROMERA (13399)

SERGIO MARTÍNEZ HAMDOUN (M19155)

Ingeniería del Software

Universidad Politécnica de Madrid

6 de enero de 2021

Contenido

[Introducción 1](#_Toc60916666)

[Propósito 1](#_Toc60916667)

[Alcance del Producto 1](#_Toc60916668)

[Realización de los requisitos en Simulink® Requirements™ 2](#_Toc60916669)

[Desarrollo 2](#_Toc60916670)

[Selección de los requisitos a evaluar mediante test automatizados 2](#_Toc60916671)

[Requisitos para testear 2](#_Toc60916672)

[Explicación de los test realizados (basados en los requisitos) 2](#_Toc60916673)

[Introducción 2](#_Toc60916674)

[Requisito cruzado con el sistema de puerta automática para gatos 2](#_Toc60916675)

[RFU10 - Identificación única y personal en la aplicación 2](#_Toc60916676)

[Conclusiones 2](#_Toc60916677)

Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Fecha** | **Razón de los Cambios** | **Versión** |
| Versión inicial | 06/01/2021 | Redacción del documento y estructura general | 1.0 |

# Introducción

## Propósito

El objetivo de este documento es doble. Por un lado, se pretende mostrar una parte de la implementación de nuestro sistema ***Güertito***. Esta implementación se realizará utilizando Matlab y Simulink, tanto en forma de diagramas de estados como funciones definidas en Matlab.

El segundo objetivo deriva del primero, puesto que con estas pequeñas implementaciones podremos realizar, y como veremos a continuación, con éxito, diversos test automatizados para mostrar el correcto funcionamiento de las partes que se han implementado.

Cabe destacar que no se ha procedido a la implementación de un modelo completo del sistema. La razón es que de esta manera, se consiguen implementar modelos más precisos y que se asemejen más a la realidad, para que una vez que se hayan desarrollado los test de forma exitosa, proceder a la implementación final en los microcontroladores.

Es importante resaltar que los test surgen para comprobar ciertos requisitos, y sobre estos requisitos se ha realizado la implementación de los modelos y los test automatizados. Por lo tanto este documento está estructurado en dos partes: primero se muestran los requisitos programados con Simulink® Requirements™ de forma general, y posteriormente se escogen algunos requisitos para probar, con sus modelos correspondientes.

## Alcance de los test

Como ya se ha dicho, el objetivo de los test es el derivado de la propia definición de test. Además, es destacable que se han ido cambiando los test para que el resultado sea satisfactorio, por lo tanto se mostrarán los resultados que han salido positivos. El nivel de detalle de los modelos deriva del cumplimiento de los requisitos que se han seleccionado para testear, siendo especialmente importante lo dicho por los requisitos para el desarrollo de los test y su resultado definitivo.

# Realización de los requisitos en Simulink® Requirements™

## Desarrollo

Para poder crear los requisitos se ha seguido un procedimiento similar al visto en la sesión de teoría de la asignatura Ingeniería del software. Se ha creado un proyecto de Matlab y en él, utilizando la herramienta Simulink® Requirements™ se ha creado una tabla donde se han ido introduciendo todos los requisitos que ya se definieron en el documento de entra anterior llamado “*Especificación de Requisitos del Sistema Güertito*”. Se puede ver un ejemplo de la estructura en la *Ilustración 1*, y un ejemplo de requisito desarrollado en la *Ilustración 2:*

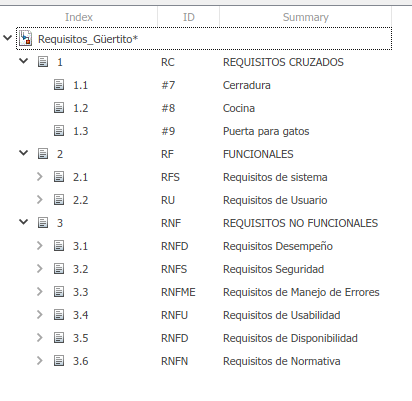


Ilustración : Estructura de los requisitos del sistema Güertito

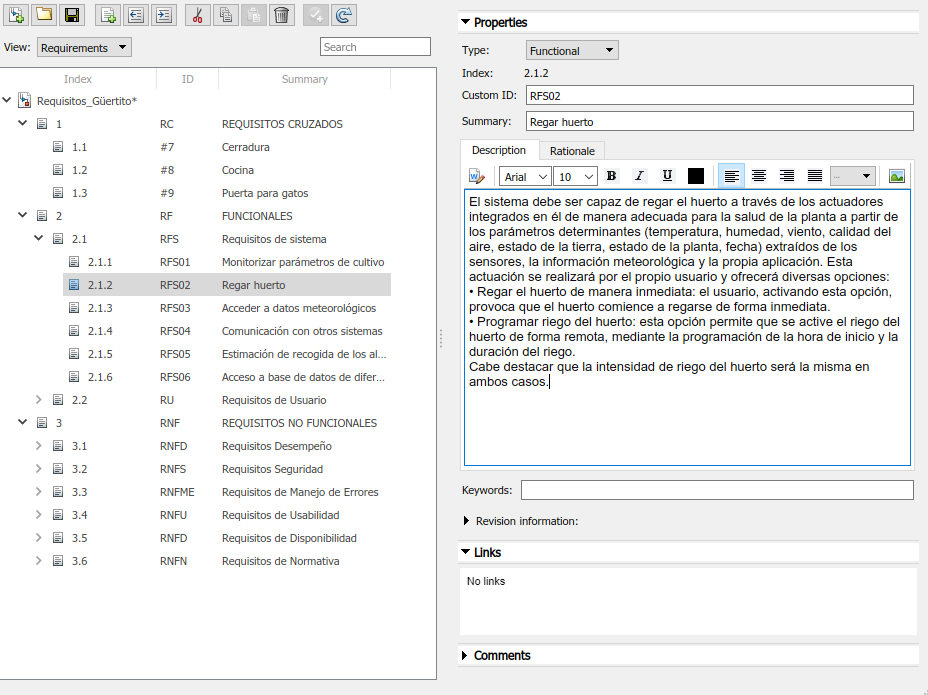


Ilustración : Ejemplo de requisito implementado del sistema Güertito

Una vez que se han implementado los requisitos, se ha procedido a seleccionar aquellos a evaluar, con la implementación de los modelos correspondientes.

# Selección de los requisitos a evaluar mediante test automatizados

## Requisitos para testear

Tabla : Tabla con los requisitos probados

|  |  |  |
| --- | --- | --- |
| CÓDIGO | TÍTULO | REQUISITO |
| Requisito cruzado | Puerta para gatos | La aplicación de la gatera se conectará por una API a los datos recopilados por el sistema del huerto acerca de la meteorología, y avisará en caso de precipitaciones para poder programar acciones como cerrar la gatera para evitar que el gato salga de la vivienda. |
| RFU10 | Identificación única y personal en la aplicación | El sistema debe proporcionar un usuario único y personal a cada persona que utilice la aplicación, debiéndose verificar la verdadera identidad de la persona que está usando un determinado usuario. El login en la aplicación se realizará mediante un sistema de verificación en dos pasos, estableciendo una clave única para cada usuario y una verificación vía SMS o correo electrónico. Como precondición se establece que el usuario esté dado de alta en el sistema y haya realizado dicha verificación. Una vez dado de alta se le asignar se podrá asignar un rol al usuario tal y como se indica en el requisito RFU11. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

\*Los códigos de los requisitos vienen explicados al final del documento, en el apéndice A: Glosario

Para cada requisito se ha desarrollado un modelo de implementación. Además, se ha establecido una relación entre los requisitos probados y los test implementados.

Tabla : Relación entre los requisitos y los números de test aplicados a cada uno de ellos

|  |  |
| --- | --- |
| REQUISITO | NÚMERO DE TEST |
| Requisito cruzado | 3 |
| RFU10 | 3 |
|  |  |

# Explicación de los test realizados (basados en los requisitos)

## Introducción

Se presentan, por lo tanto, los modelos de implementación de cada requisito, con la explicación de los test que se han llevado a cabo sobre cada modelo. El hecho de incluir varios test dentro de cada modelo se basa en mostrar el comportamiento distinto del sistema ante un cambio de entradas. Todo esto se mostrará por el cambio de las señales de salida.

## Requisito cruzado con el sistema de puerta automática para gatos

El modelo de implementación que conecte nuestro sistema con el sistema de apertura y cierre automático para gatos se basa en que, cuando se detecten precipitaciones cercanas, se genere una señal de alarma que se envíe a dicho sistema. Por lo tanto lo que se ha implementado es una secuencia mediante la cual se accedan a los datos meteorológicos. Si se ha accedido de forma correcta a la información meteorológica, se genera un pulso de salida en la señal “*pronóstico*”. El siguiente paso consiste en acceder a la información de precipitaciones, sólo si se ha accedido de forma correcta al pronóstico temporal. Si se obtiene la información de precipitaciones, se pasa a evaluar si son cercanas o no, parámetro que se determinar por la señal “*lluvia*”, que sigue la siguiente lógica:

*if (lluvia==1)*

*Alarma*

*if (lluvia == 2)*

*No\_alarma*

La señal de alarma o no alarma viene definida por “*mensaje\_gatera”,* que envía esta información codificada al sistema de la puerta automática para gatos, para que posteriormente este sistema actúe en consecuencia de esta información enviada.

Se ha implementado un sistema de fallos, de tal forma que si no hay conexión a internet, se incrementen el número de intentos que de forma automática el sistema acceda a internet para recopilar los datos meteorológicos, y así se mande un mensaje de error a la aplicación del usuario del sistema Güertito, para que lo vuelva a intentar. Esta señal viene implementada como “error”.

Por lo tanto, el sistema modelado derivado de este requisito tiene las siguientes señales de entrada y de salida:

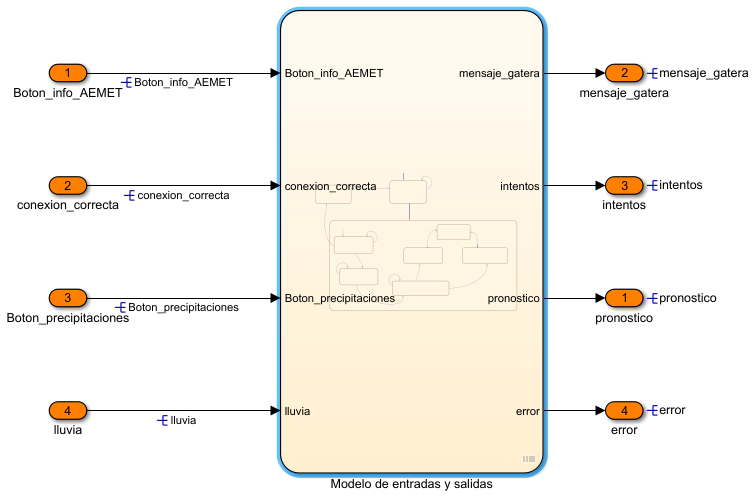


Ilustración : Modelo de entradas y salidas

Se ha intentado modelar de alguna forma los botones que el usuario tiene que pulsar para acceder a la información de las precipitaciones, que son “*Boton\_info\_AEMET”* y “*Boton\_precipitaciones”.* Además, se ha creado la señal de “*intentos*” simplemente con el objetivo de monitorizar los intentos que el sistema realiza para acceder de forma correcta a la información del tiempo. Por lo tanto, el modelo que resulta, teniendo en cuenta todo lo anterior, es el siguiente:

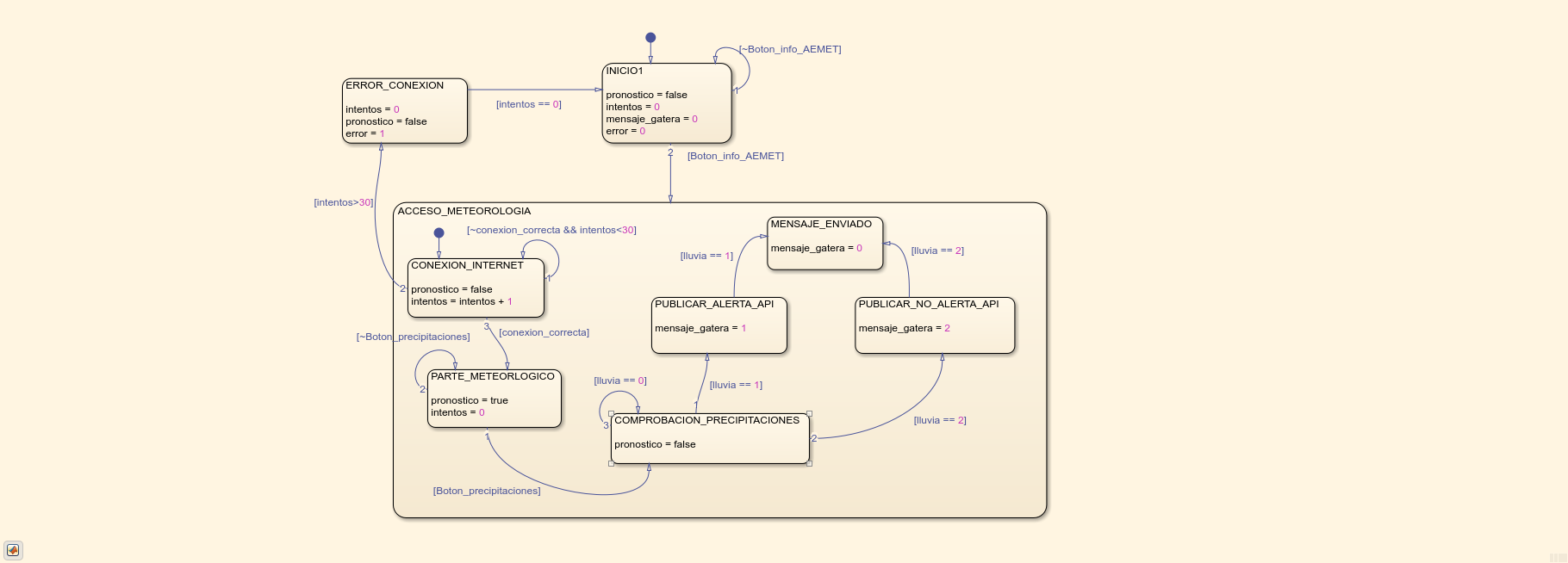


Ilustración : Modelo de conexión con el sistema de la puerta de gatos que genera una alerta de lluvias

## RFU10 - Identificación única y personal en la aplicación

# Conclusiones

# Apéndices

**Apéndice A: Glosario**

|  |  |
| --- | --- |
| **RFS** | **Requisito Funcional de Sistema** |
| **RFU** | **Requisito Funcional de Usuario** |
| **RNF** | **Requisito No Funcional** |
| **RNFD** | **Requisito No Funcional Desempeño** |
| **RNFS** | **Requisito No Funcional Seguridad** |
| **RNFME** | **Requisito No Funcional Manejo de Errores** |
| **RNFU** | **Requisito No Funcional de Usabilidad** |
| **RNFDI** | **Requisito No Funcional de Disponibilidad** |
| **RF** | **Requisito Funcional** |
| **Cliente** | **Cualquier tipo de usuario de los descritos en la sección Clases de Usuario y Características.** |
| **RNFN** | **Requisito No Funcional de Normativa** |