PROGRAMACION ORIENTADA A OBJETOS

App de control de casa

Semestre 2011-2012/II

[No de proyecto](#h.z8dwovkpuhl8)

[Nombre del proyecto](#h.mnxc81u1z897)

[Integrantes del equipo](#h.dntok8juv9p3)

[Objetivo del proyecto](#h.bsu9evf8nxs1)

[Descripción del proyecto](#h.eivcmejh9ing)

[Descripción e imágenes de cada nivel](#h.dipcv9h52uj1)

Portada

Ayuda

Créditos

Interface

Radio

Televisión

Luz

Microondas

[Diagrama de clases UML](#h.g0vt53pc7r37)

[Características y comportamiento de cada clase](#h.mtuw3jfplyj6)

[Herencia y polimorfismo](#h.k1a372vuic1q)

[Cronograma de actividades (plan de trabajo)](#h.30zz6o5p35xo)

[Bitácora de actividades (historial)](#h.rr6kuez6yf7u)

No de proyecto

El número asignadado al proyecto conforme al sistema de registros en el portal del Área de computación e informática es el número 101

1. Nombre del proyecto
2. App de control de casa
3. Integrantes del equipo

248818 Pilar Carmona Diego

1. Objetivo del proyecto

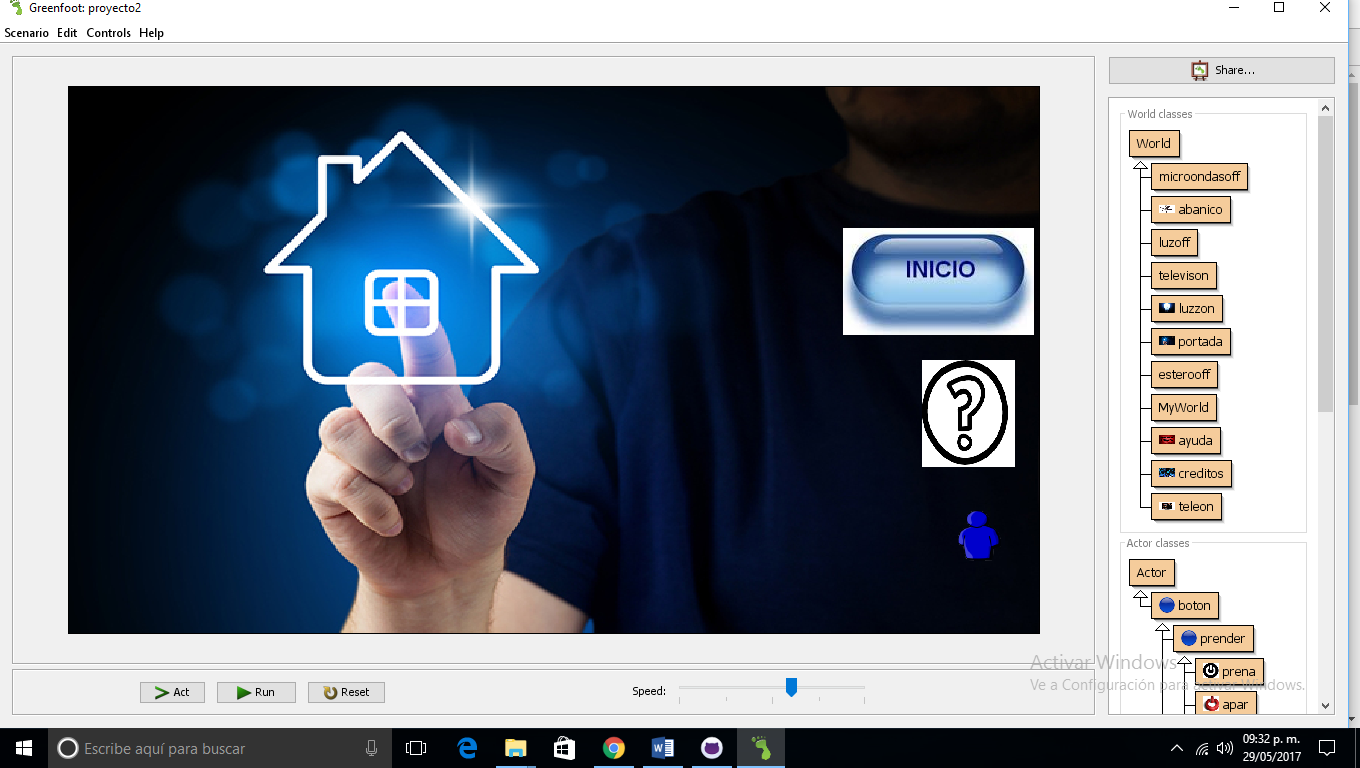
El objetivo consiste en crear una aplicación que simula el control de varios aparatos de una casa

1. Descripción del proyecto
2. La descripción del proyecto consiste en crear una aplicación que simule los diferentes aparatos

Que se pueden controlar, una televisión y emula el prendido de la tele y el cambiado de canal, la luz, el escenario cambia, el microondas simula el ruido de un microondas entre otros aparatos más.

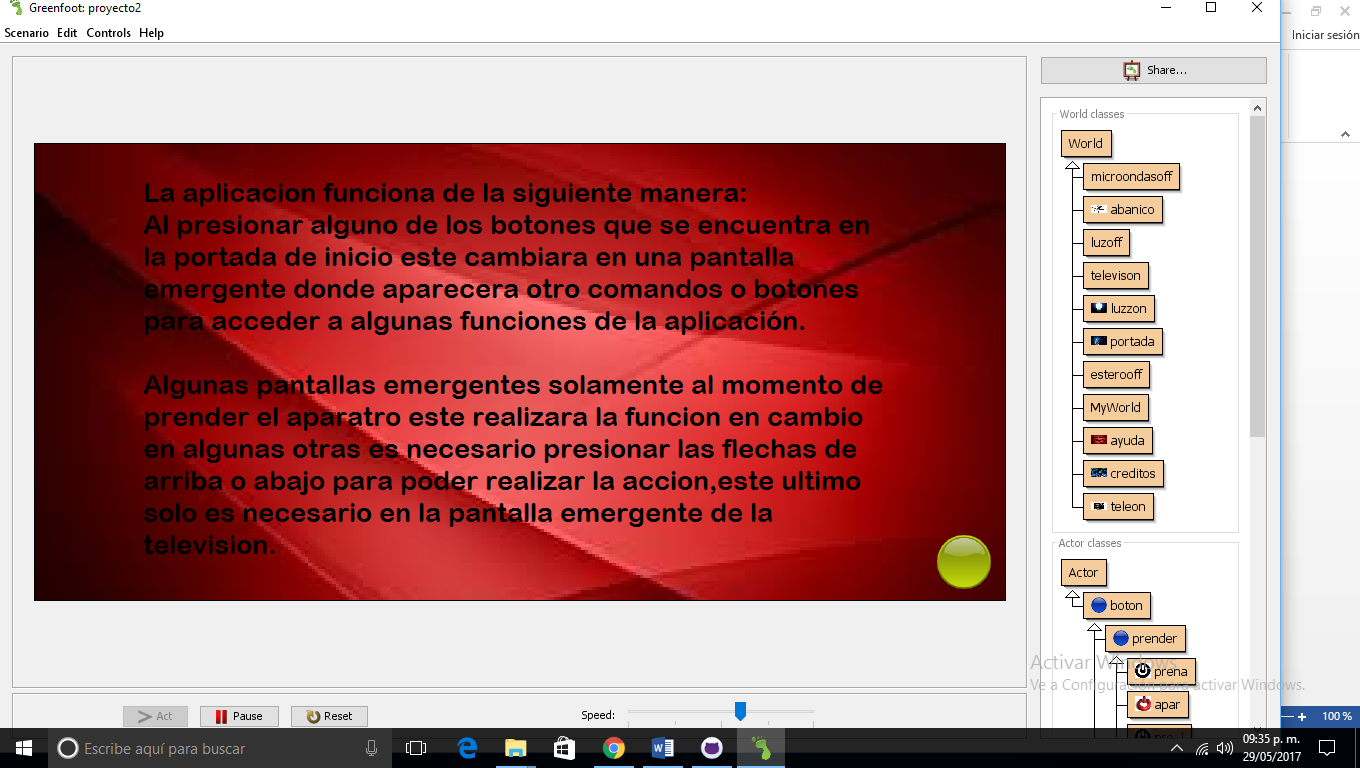
1. Descripción e imágenes de cada nivel

* Portada



En la portada se encuentras 3 opciones que permiten movernos a la ayuda, créditos y a la interface de la aplicación.

* Ayuda



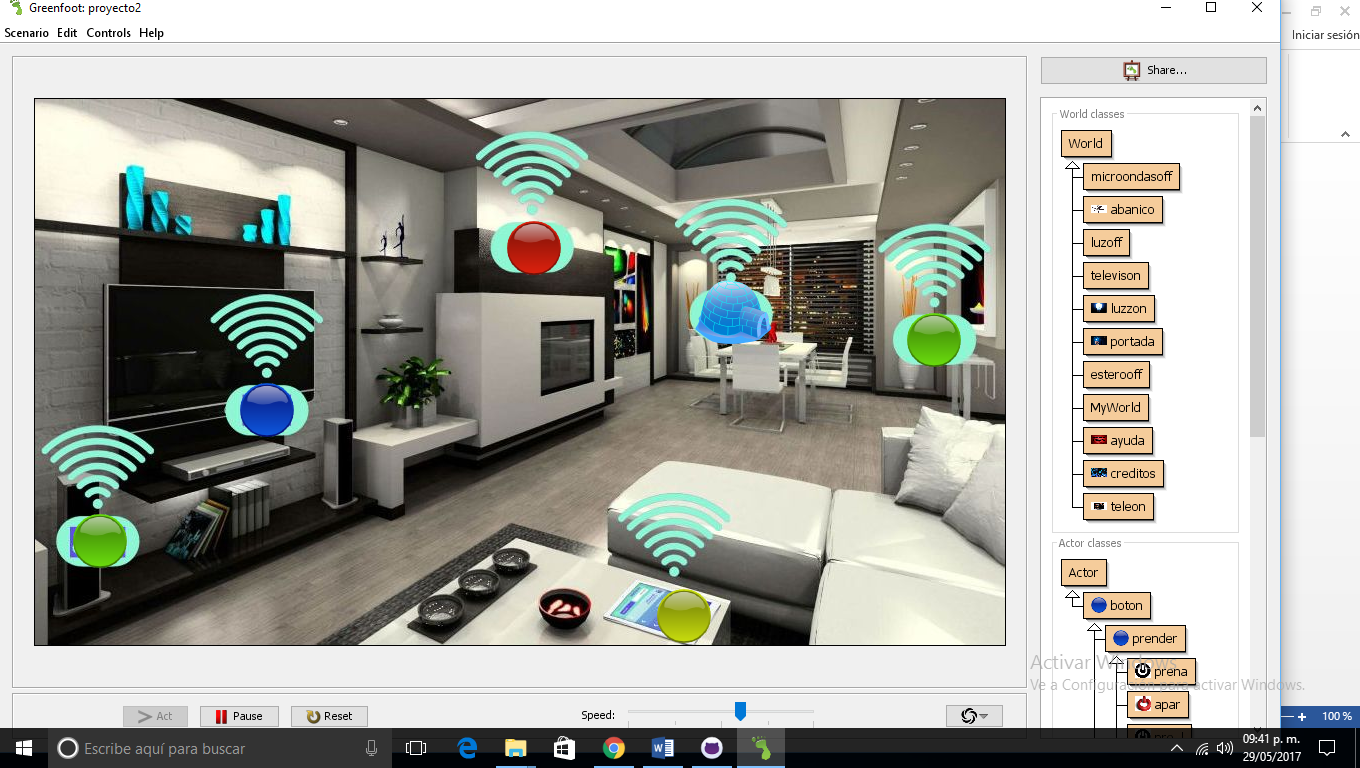
Al presionar la tecla de ayuda se mostrara cómo funciona la aplicación y algunas otras indicaciones.

* Créditos



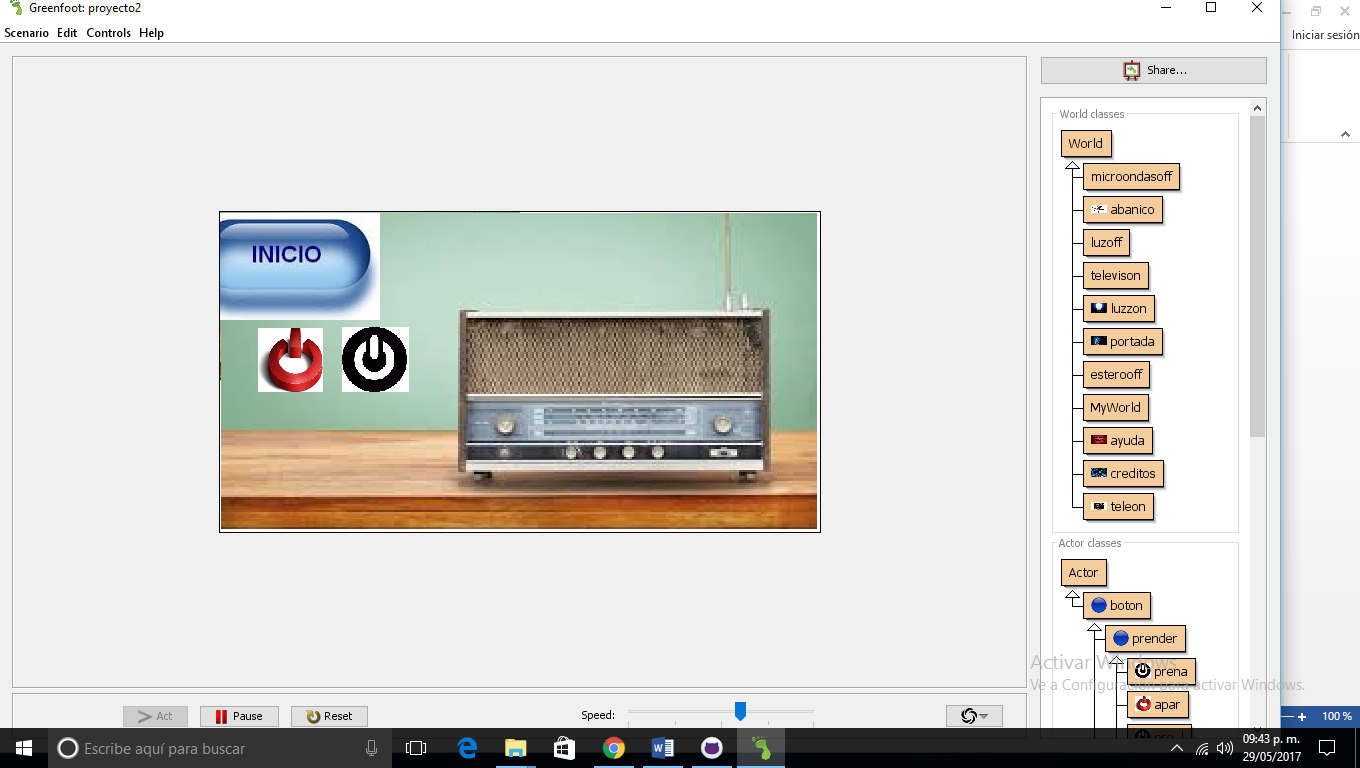
Aquí se muestra el profesor que imparte la materia y quien realizo la app.

* Interface



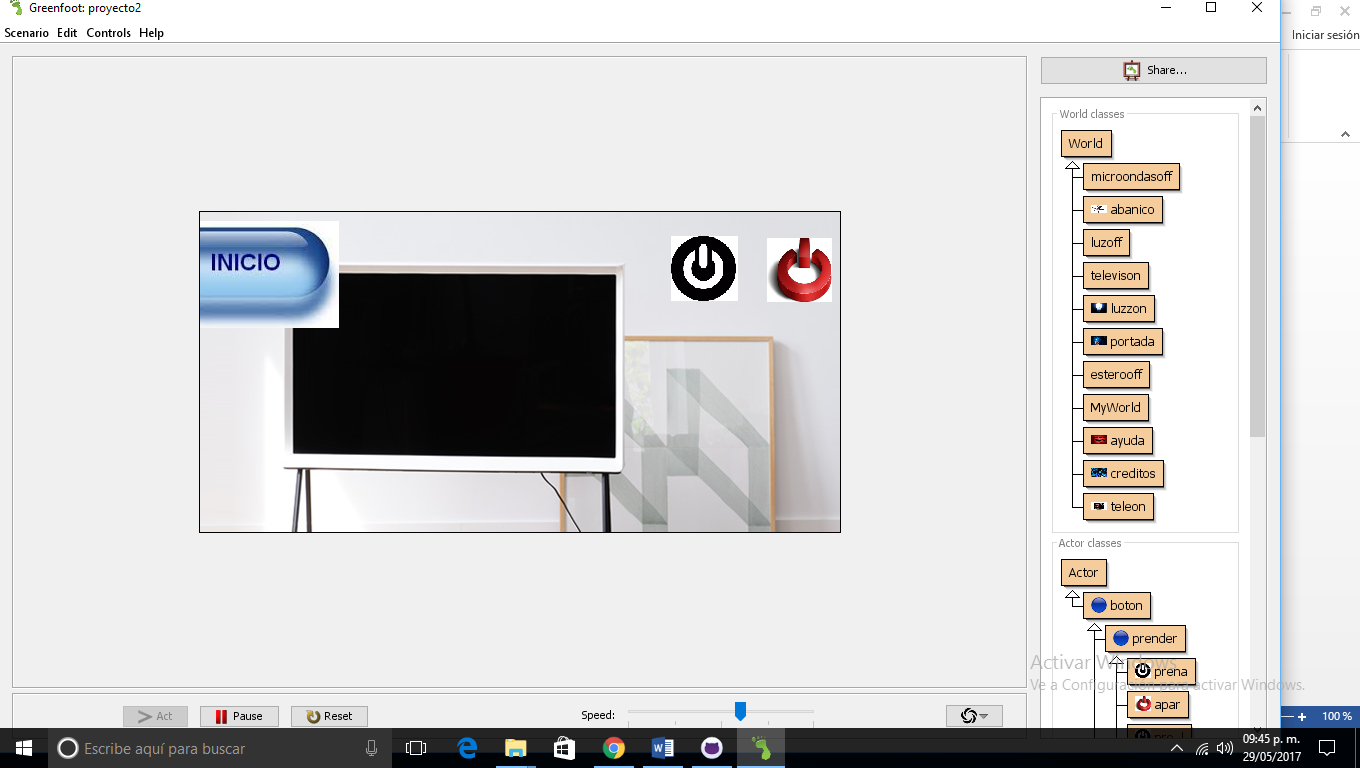
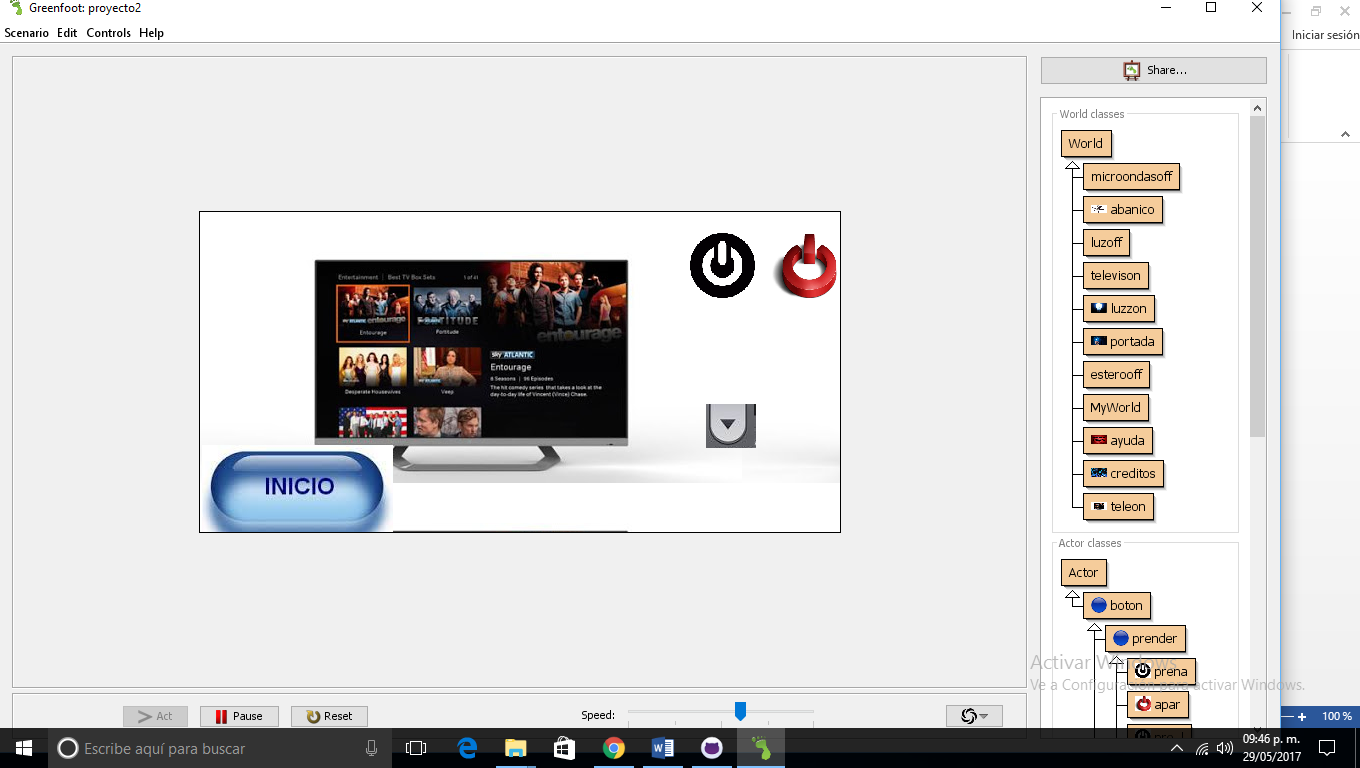
Se muestra la interface de la app, y funciona de la manera que al presionar algún botón este nos abrirá una ventana emergente que simula que ha accedido al control de dicho objeto.

* Radio



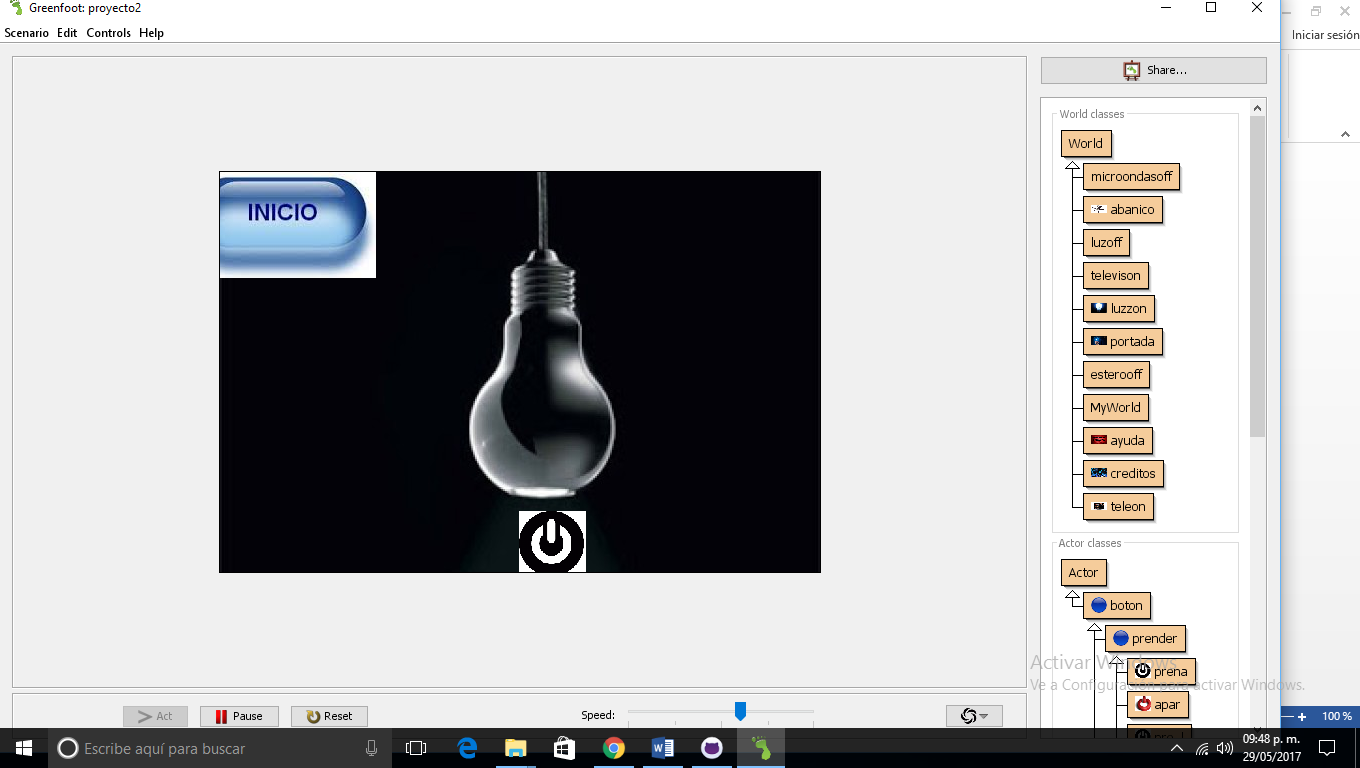
En esta ventana al momento de presionar el botón de encendido de la radio este automáticamente emite el ruido de la radio simulando el encendido del aparato.

* Televisión

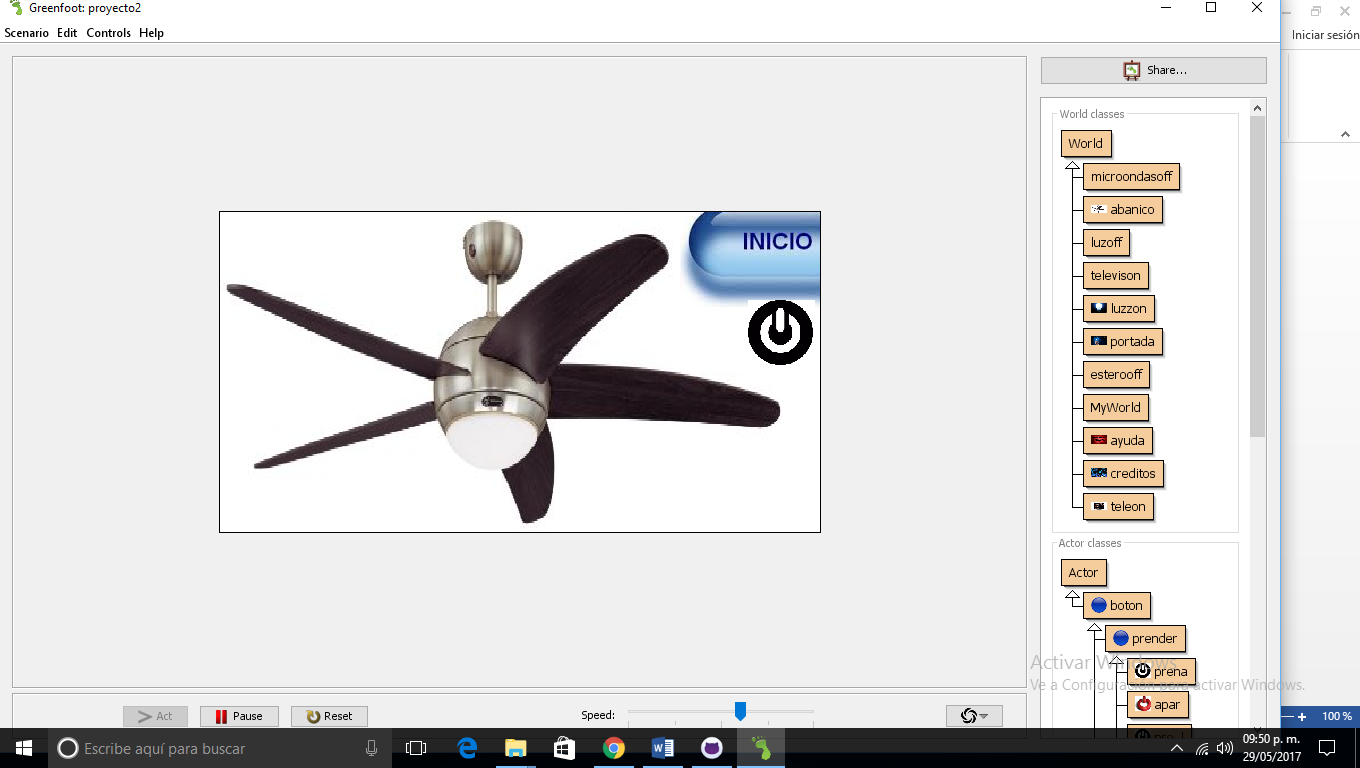
Al momento de presionar el boton de encendido este realiza el cambio de escenario y esta a la espera de que se presione el boton de flecha arriba y flecha abajo para cambiar de canal.

* Luz

Al presionar el boton de la luz al presionar el boton de encendido este cambia de escenario y simula el ruido de prenido de la luz.

* Abanico



Al presionar el botón de encendido del abanico simula el ruido que hace al presionar el botón de encendido del abanico

* Microondas



Este al igual que el abanico este al presionar el botón de encendido simula el ruido que haría como si el microondas estuviera actuando en ese momento.

1. Diagrama de clases UML

Botón

Prender

Canal

1. Características y comportamiento de cada clase

Escribir aquí las características de las principales clases de su proyecto.

|  |  |
| --- | --- |
| Nombre de la clase: | Botón |
| Características: | Las características principales de esta clase sirven para poder realizar el cambio de escenario de cualquier aparato de la casa. |
| Comportamiento: | Su comportamiento principal consiste en es padre de varias clases y estas sus métodos son verificar si son presionadas y si es así llaman a la función de esta clase y dependiendo que objeto fue presionado este envía un valor y realiza el cambio de escenario |

*\* Para añadir más renglones a la tabla primero debe seleccionar el renglón y después desde el menú “****Tabla****” seleccionar la opción “****Insertar una fila debajo****”*.

|  |  |
| --- | --- |
| Nombre de la clase: | Prender |
| Características: | La característica principal de esta clase es que realiza el activado del sonido de los objetos o aparatos. |
| Comportamiento: | El comportamiento es cuando se realiza que simule la actividad del aparato y este detiene o reproduce el sonido deseado para poder crear la ilusión que se realiza la app. |

*\* Para añadir más renglones a la tabla primero debe seleccionar el renglón y después desde el menú “****Tabla****” seleccionar la opción “****Insertar una fila debajo****”*

1. Herencia y polimorfismo

La herencia que se aplicó en este proyecto fue en la creación de botón debido a que todos los demás objetos creadas a partir de esta clase heredan una función principal que realiza el cambio de escenario dependiendo el valor que se le da por parámetro a dicho función.

1. Cronograma de actividades (plan de trabajo)

Describir todas las actividades por realizar desde la propuesta del proyecto hasta la entrega considerando los siguientes entregables:

* + Manual del usuario
  + Manual del programador (este documento)
  + Código
  + Video
  + Link a Greenfoot
  + Link a google code

Se deben tomar en cuenta las siguientes fechas importantes:

* + fecha de entrega para examen de ordinario: 31 de mayo
  + fecha de entrega para examen de extraordinario: 2 de junio

|  |  |  |
| --- | --- | --- |
| Fecha de Inicio | Fecha de Término | Actividad por realizar |
| 11-04-17 | 13-04-17 | Planeación del diagrama |
| 8-05-17 | 20-05-17 | Creación de la aplicación |
| 27-05-17 | 29-05-17 | Creación del ejecutable |
| 29-05-17 | 31-05-17 | Creación de la documentación necesaria |
| 29-05-17 | 31-05-17 | Creación del video |
| 29-05-17 | 31-05-17 | Subir el juego a la página principal de greenfoot |

*\* Para añadir más renglones a la tabla debe seleccionar primero la tabla y después desde el menú “****Tabla****” seleccionar la opción “****Insertar una fila debajo****”*

1. Bitácora de actividades (historial)

Esta parte será llenada durante la elaboración del proyecto. Aquí se deben describir cada una de las actividades realizadas desde la propuesta hasta la entrega del proyecto.

|  |  |
| --- | --- |
| Fecha (dd/mm/aa) | Descripción de la actividad realizada |
| 11-04-17 | Creación del primer borrador del diagrama de clases |
| 12-04-17 | Verificación y corrección del diagrama |
| 13-04-17 | Diagrama final de clases del proyecto |
| 27-05-17 | Creación del proyecto crear la primera etapa |
| 28-05-17 | Seguimiento del proyecto |
| 29-05-17 | Terminación del proyecto |
| 30-05-17 | Aclaración de detalles finales y empezar la documentación |
| 30-05-17 | Subir los archivos a github |
| 30-05-17 | Creación del ejecutable |
| 30-05-17 | Creación del juego en página web |