**Universidad ORT Uruguay**

**Facultad de Ingeniería**

**Bernard Wand Polak**

**Diseño de Aplicaciones**

**Obligatorio 1**



Emily Symonds Diego Carrión

172168 172129

2014

Tabla de contenido

[1. Introducción 4](#_Toc404004168)

[2. Justificación de Diseño 4](#_Toc404004169)

[2.1. Patrones utilizados: 4](#_Toc404004170)

[2.1.1. Singleton: 4](#_Toc404004171)

[2.1.2. Composite: 4](#_Toc404004172)

[2.1.3. Observer 5](#_Toc404004173)

[2.2. Separación de proyectos: 5](#_Toc404004174)

[2. Descripción de las principales decisiones de diseño tomadas 6](#_Toc404004175)

[3. Co mentarios 7](#_Toc404004176)

[3.1. Sobre diagramas de clases 7](#_Toc404004180)

[3.2. Consideraciones del diseño y del código 7](#_Toc404004181)

[4. Diagramas de paquetes 8](#_Toc404004182)

[5. Diagramas de clases (Ver con más claridad en directorio) 9](#_Toc404004183)

[5.1. UAVInterfaz 9](#_Toc404004184)

[5.2. UAVLógica 10](#_Toc404004185)

[5.3. Diagrama entre paquetes 11](#_Toc404004186)

[6. Diagramas de Interacción 12](#_Toc404004187)

[6.1. Actualizar valor de un sensor utlizando el observer 12](#_Toc404004188)

[6.2. Alarma urgente excedida. Destrucción del dron. 13](#_Toc404004189)

[6.3. Agregar Diseño 14](#_Toc404004190)

[7. Casos de uso 15](#_Toc404004191)

[7.1. Agregar una pieza 15](#_Toc404004192)

[7.2 Editar una pieza 16](#_Toc404004193)

[7.3 Eliminar una pieza 16](#_Toc404004194)

[7.4 Agregar un componente reusable 17](#_Toc404004195)

[7.5 Editar un componente reusable 18](#_Toc404004196)

[7.6 Eliminar un componente reusable 19](#_Toc404004197)

[7.7 Ver un componente reusable 20](#_Toc404004198)

[7.8 Agregar un diseño 20](#_Toc404004199)

[7.9 Editar un diseño 22](#_Toc404004200)

[7.10 Agregar un dron 23](#_Toc404004201)

[7.11 Editar un dron 24](#_Toc404004202)

[7.12 Eliminar un dron 24](#_Toc404004203)

[7.13 Inicar una simulación 25](#_Toc404004204)

[Glosario: 27](#_Toc404004205)

# Introducción

UAV es una aplicación que maneja la creación y simulación de drones. Permite la creación de piezas para el mismo, y componentes reusables compuestos por piezas u otros componentes. Luego de crear las piezas que componen un dron, se crea un diseño para el mismo.

El sistema permite realizar simulaciones de vuelo para cada dron, y de esta forma evaluar su comportamiento de acuerdo a las alarmas que han sido seteadas para cada sensor del mismo. La simulación termina si el dron llega al destino predefinido, o si en su transcurso uno de sus sensores en estado de alarma urgente superó los 5 segundos. Si ocurre lo último, el dron se destruye.

Se genera un reporte al finalizar cada simluación, y a su vez dos archivos en formato XML y CSV respectivamente, donde se registran los valores de los sensores cada cierto tiempo.

Los datos de las piezas, componentes reusables y diseños de drones son persistidos en una base de datos.

# Justificación de Diseño

## Patrones utilizados:

### Singleton:

Utilizamos el patrón Singleton para prevenir el acceso a la creación de objetos más allá de la manera en que decidimos hacerlo nosotros, es decir para evitar crear varias instancias de una clase y de esta forma evitar inconsistencias.

Se implementó este patrón para la clase Sistema (haciendo los constructores privados), logrando un acceso controlando a la única instancia, puesto que la clase Singleton (en nuestro caso Sistema), encapsula su única instancia, lo que permite tener un control estricto sobre cómo y cuándo acceden a ella los clientes.

También lo implementamos a la hora de persistir los datos, ya que para acceder a la base de datos hay que realizar una conexión a la misma. Dicha conexión debe ser única en el transucurso de la ejecución de la aplicación, ya que se generarían inconsistencias en los datos en la base de datos si no fuese de esta forma.

### Composite:

Este patrón nos facilitó la estructura de las clases Pieza y Componente, ya que identificamos que la diferencia entre éstas (compuesto y componente), en cuanto al comportamiento, es poca o prácticamente nula. Este patrón nos permitió trabajar los componentes de modo uniforme, independientemente de sus complejidades.

Suponemos que un futuro, nos permitirá agregar nuevos componentes sin mayores cambios.

### Observer

La utilidad de este patrón radica en definir una relación entre objetos de forma que cuando un objeto cambia su estado, los dependientes son avisados y se actualizan automáticamente.

En nuestro caso, nos fue útil para:

* Actualizar la consola del Piloto ante una variación en los valores de los sensores (piezas) pertenecientes al Dron durante la simulación.
* Actualizar la consola del Piloto cuando se activa la alarma, e indicando además cuando el Dron cruza el umbral mínimo o máximo de cualquiera de sus piezas (alarma urgente).

En estos casos la que notifica sobre su cambio de estado (en particular la activación de la alarma) es la clase Sistema. Quien se entera de estos cambios y se actualiza es la consola del Piloto.

Un sujeto (u observado) puede tener cualquier número de observadores dependientes de él. Cada vez que el sujeto cambia su estado se notifica a todos los observadores. Por su parte, cada observador consultará al sujeto para sincronizar su estado con el estado de éste.

## Separación de proyectos:

La razón por la cual separamos los proyectos entre UAVLogica(Dominio) e UAVInterfaz (Interfaz Gráfica), fue para que, en caso de que haya que hacer una modificación en alguno de estos proyectos, el otro no se vea alterado o sufra el menor impacto posible.

Además nos sirve para separar la lógica del programa de la interfaz de usuario, lo cual posibilita minimizar la propagación de cambios o posibles errores. Permite que el sistema sea adaptable a otras implementaciones. Por esta razón es que cumple con el Principio de Separación de la Interfaz, ya que puede utilizarse otra interfaz para presentar la solución. Intentamos que las ventanas fueran lo más simples e intuitivas posibles.

Para implementar la persistencia de datos, se crearon dos paquetes, Persistencia y PersistenciaImp.

# Descripción de las principales decisiones de diseño tomadas

Como se expuso en el punto anterior, nosotros tenemos definidos cuatro proyectos, los cuales contienen todo el tema de lógica (UAVLogica), el tema de interfaz de usuario (UAVInterfaz), y la Persistencia (Persistencia y PersistenciaImp).

En el proyecto de UAVLogica, las principales clases son:

* Sistema: la cual se encarga de tener la lista de todos los objetos creados formando la estructura del manejo de drones, así como también es quien controla dicha estructura mediante diversos métodos.
* Simulación: es quien se encarga de realizar la simulación de los drones, y permite al usuario cambiar los diferentes valores de la condición de vuelo.
* Dron: permite la creación de drones, mediante la selección de un diseño y demás atributos.
* Nodo, Pieza y Componente que son las que cumplen con el patron Composite. Pieza y Componente heredan de Nodo (compartiendo los atributos marca y modelo), y Componente está formado por Nodos (es decir Piezas y/o Componentes).
* Altímetro, Barómetro, GPS, Velocímetro, Termómetro, Acelerómetro que son los tipos de piezas existentes, y heredan de Pieza.
* Diseño: maneja los datos de los diseños para los drones.
* Archivo, ArchivoXML y ArchivoCSV que permiten la creación y actualización de los archivos generados para cada simulación. Archivo es una clase abstracta de la cual heredan ArchivoXML y ArchivoCSV.

En el proyecto de la UAVInterfaz, las principales clases son:

* MenuPpal: el menú principal permite al usuario seleccionar entre las distintas opciones tales como administrar piezas, componentes, diseños, drones, etc. Así como también realizar las simulaciones sobre los drones.
* Reporte: muestra información acerca de los de valores de las condiciones de vuelo de una simulación cada 30 segundos.
* ConsolaAdministrativa: permite al usuario modificar los valores de las condiciones de vuelo (velocidad, temperatura, presión, etc).
* Piloto: esta ventana muestra, para la simulación en curso en el sistema, los nuevos valores actualizados de dicha simulación e indica cuando se disparan las alarmas para cada una de las piezas correspondientes.
* Por último existen ventanas para el alta, baja o modificación de datos como: piezas, componentes, alarmas, diseños y drones.

# Co mentarios



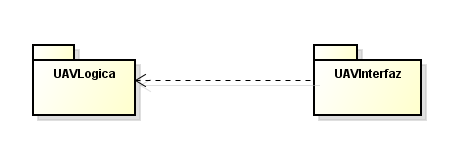
## Sobre diagramas de clases

Debido a la magnitud de los proyectos Dominio e Interfaz, en esta documentación se publicarán solamente los principales diagramas pero no se podrán observar los detalles de las relaciones de clase. Para ello, favor de mirar dichos diagramas en el documento de nombre **Diagramas** ubicado en el directorio**.**

## Consideraciones del diseño y del código

1. Los identificadores (varios de uso interno) son autogenerados. El usuario puede modificarlos en varios casos pero el sistema controla que no se duplique uno existente.
2. La destrucción del Dron consiste en disminuir la cantidad de dicho Dron (le restamos uno al atributo cantidad).
3. Para saber si una pieza está en estado de alarma, revisamos que el valor actual más la precisión o que el valor actual menos la precisión no supere o sea menor al valor de la alarma definido.
4. Lo mismo para con la alarma urgente (valor actual + precisión >= máximo y valor actual – precision <= mínimo).
5. Para el caso del Velocímetro, Acelerómetro y GPS, los cuales cuentan con tres valores (x, y, z), definimos un único valor de mínimo y máximo para las tres coordenadas, así como un valor único de alarma.
6. A la hora de editar los datos de una pieza, no permitimos la modificación del tipo de pieza.
7. Sistema notifica (IObservable) a la ventana Piloto (IObserver) sobre su cambio de estado.
8. Tener en cuenta que los datos de prueba son ficticios.
9. En el caso de la simulación automática, generamos datos de prueba cada 4 segundos, los cuales van cambiando automáticamente las condiciones del vuelo hasta finalizarlo.
10. Los archivos XML y CSV se crean al inicar la simulación, y son actualizados cada vez que se modifica algún sensor y también en el Tick del timer cada 30 segundos.
11. En los campos a ingresar, corroboramos que los mismos no sean ingresados vacíos, pero no validamos el tipo de dato ingresado, ya sea decimal, entero o string.
12. A la hora de crear un Diseño, obligamos a que el mismo contenga al menos un GPS y un Velocímetro, ya que consideramos estas piezas indispensables.

# Diagramas de paquetes



# interfaz.PNGDiagramas de clases (Ver con más claridad en directorio)

# UAVInterfaz

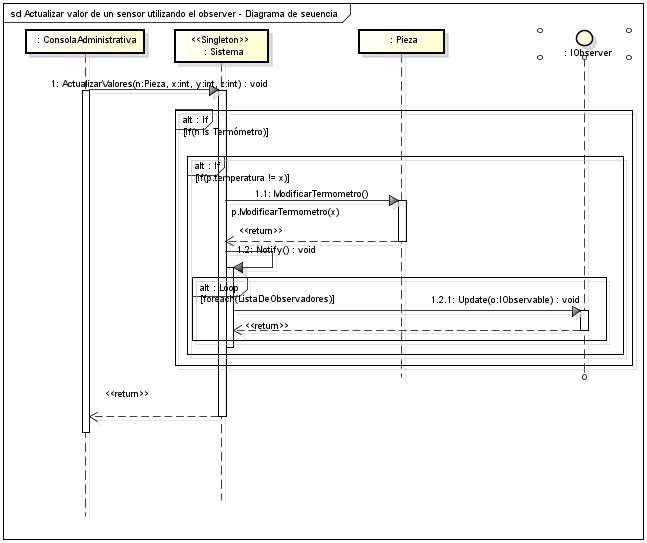
# logica.PNG UAVLógica

# d.PNG Diagrama entre paquetes

# Diagramas de Interacción

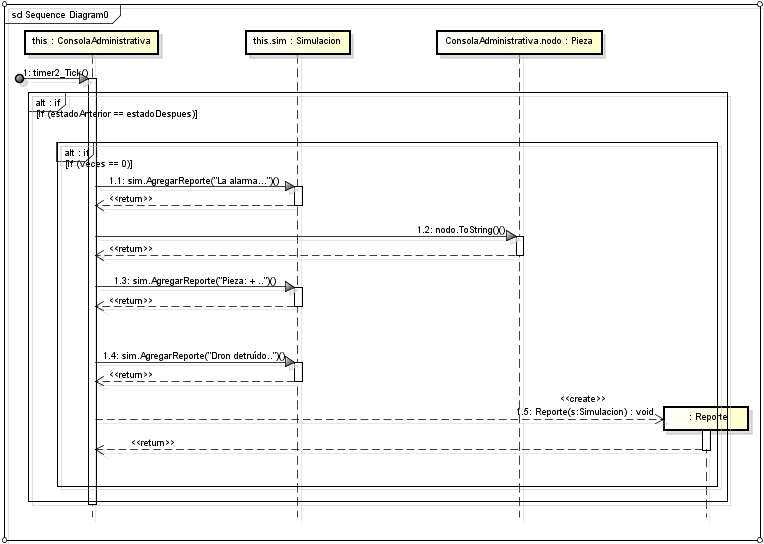
# Actualizar valor de un sensor utlizando el observer

Desde la Consola Administrativa se actualiza el valor de una Pieza llamado al método ActualizarValores en Sistema. En el Sistema, de acuerdo a que tipo de Pieza se recibe en el método (Termómetro en el ejemplo), llama al método en la Pieza que modifica el valor de su sensor (ModificarTermometro()). Luego de modificar se llama al Notify() desde el Sistema, que hace un Update en la Lista de Observadores, actualizando el valor del sensor. El Update() es un método que hereda de Iobserver.



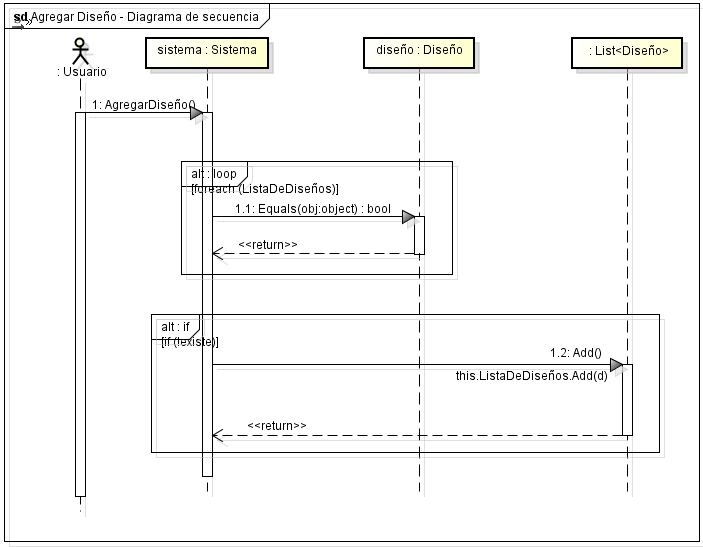
# Alarma urgente excedida. Destrucción del dron.

En la Consola Administrativa se inicia un Timer de 5 segundos cuando un sensor dispara una alarma urgente. Si luego de 5 segundos el estado de la alarma sigue en estado urgente, se debe eliminar el dron (restándole 1 a la cantidad del mismo), por lo que se agrega al Reporte la razón de la destrucción y qué pieza la causó.



# Agregar Diseño

Desde la Interfaz Agregar Diseño, se llama al método en Sistema AgregarDiseño(). Verifica que no exista un diseño con el mismo nombre, y en caso negativo lo agregar a la Lista de Diseños del sistema.



# Casos de uso

# Agregar una pieza

|  |  |
| --- | --- |
| **Descripción:** El usuario agrega una pieza | |
| **Actores:** Usuario | |
| **Precondición:** - | |
| **Postcondición:** Se agrega una pieza | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. Selecciona el ítem Componentes de vuelo en la pantalla principal |  |
|  | 2.Despliega un submenú con los ítems Administar Piezas y Administrar Componentes Reusables. |
| 3. Selecciona el ítem Administar Piezas |  |
|  | 4. Muestra un listado de todas las piezas ingresadas hasta el momento y opciones para agregar, modificar o eliminar una pieza. |
| 5. Selecciona el botón para Agregar una pieza. |  |
|  | 6. Despliega una ventana para completar los datos de la pieza. |
| 7. Ingresa los datos de la pieza y selecciona el botón Agregar. |  |
|  | 1. Registra la pieza, y muestra un mensaje de confirmación. Vuelve a la ventana de Administración de piezas. |
| **Curso Alternativo 1** | |
| 1. Ingresa los datos excepto un campo y selecciona el botón Agregar. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |
| **Curso Alternativo 2** | |
| 1. Ingresa los datos, donde la marca y el modelo ingresados ya existen en otra pieza, y selecciona el botón Agregar. |  |
|  | 1. Muestra un mensaje de error indicando que ya existe una pieza con esa marca y modelo. |
| **Curso Alternativo 3** | |
| 13. Hace click en el botón **Volver.** |  |
|  | 14. El sistema volverá a la ventana de Administración de piezas. |

## 

# Editar una pieza

|  |  |
| --- | --- |
| **Descripción:** El usuario edita una pieza existente | |
| **Actores:** Usuario | |
| **Precondición:** La pieza debe estar ingresada en el sistema | |
| **Postcondición:** Modifica una pieza | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de piezas, selecciona una pieza y hace click en el botón Editar |  |
|  | 2. Despliega una ventana donde se permite modificar los campos para los datos de la pieza excepto el modelo, la marca y el tipo de pieza. |
| 3. Modifica alguno de los datos disponibles y hace click en Editar. |  |
|  | 4. Muestra un mensaje de confirmación de la modificación. |
| **Curso Alternativo 1** | |
| 1. Ingresa los datos excepto un campo y selecciona el botón Editar. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |
| **Curso Alternativo 2** | |
| 1. Hace click en el botón **Volver** |  |
|  | 1. El sistema volverá a la ventana de Administración de piezas. |

# Eliminar una pieza

|  |  |
| --- | --- |
| **Descripción:** El usuario elimina una pieza existente | |
| **Actores:** Usuario | |
| **Precondición:** La pieza debe estar ingresada en el sistema | |
| **Postcondición:** Elimina una pieza | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de piezas, selecciona una pieza y hace click en el botón Eliminar. |  |
|  | 1. Le pregunta al usuario si está seguro de querer eliminar la pieza. |
| 1. Selecciona la opción afirmativa. |  |
|  | 1. Elimina la pieza y muestra un mensaje de confirmación. Vuelve a la ventana de Administración de piezas. |
| **Curso Alternativo 1** | |
| 1. Selecciona la opción afirmativa. |  |
|  | 1. No elimina la pieza ya que forma parte de algún componente o diseño por lo que muestra un mensaje de error |
| **Curso Alternativo 2** | |
| 1. Selecciona la opción negativa. |  |
|  | 1. Vuelve a la ventana de Administración de piezas. |

# Agregar un componente reusable

|  |  |
| --- | --- |
| **Descripción:** El usuario agrega un componente reusable | |
| **Actores:** Usuario | |
| **Precondición:** Que hayan piezas ingresadas previamente | |
| **Postcondición:** Se agrega un componente reusable | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. Selecciona el ítem Componentes de vuelo en la pantalla principal. |  |
|  | 1. Despliega un submenú con los ítems Administar Piezas y Administrar Componentes Reusables. |
| 1. Selecciona el ítem Administar Componentes Reusables. |  |
|  | 1. Muestra un listado de los componentes ingresados hasta el momento y opciones para ver, agregar, modificar o eliminar un componente. |
| 1. Selecciona el botón para Agregar un componente. |  |
|  | 1. Despliega una ventana para ingresar los datos del componente y la selección de piezas u otros componentes que lo pueden conformar. |
| 1. Ingresa los datos del componente, selecciona las piezas y/o componentes que lo forman y hace click en el botón Crear. |  |
|  | 1. Registra el componente, y muestra un mensaje de confirmación. Vuelve a la ventana de Administración de componentes. |
| **Curso Alternativo 1** | |
| 1. Ingresa los datos excepto un campo y selecciona el botón Crear. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |
| **Curso Alternativo 2** | |
| 1. Ingresa los datos, donde la marca y el modelo ingresados ya existen en otra pieza, y selecciona el botón Crear. |  |
|  | 1. Muestra un mensaje de error indicando que ya existe una pieza con esa marca y modelo. |
| **Curso Alternativo 3** | |
| 1. Ingresa los datos pero no selecciona ninguna pieza o componente y selecciona el botón Crear. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |
| **Curso Alternativo 4** | |
| 15. Hace click en el botón **Volver.** |  |
|  | 16. El sistema volverá a la ventana de Administración de piezas. |

# Editar un componente reusable

|  |  |
| --- | --- |
| **Descripción:** El usuario edita un componente existente | |
| **Actores:** Usuario | |
| **Precondición:** El componente debe estar ingresado en el sistema | |
| **Postcondición:** Modifica un componente | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de omponentes, selecciona un componente y hace click en el botón Editar. |  |
|  | 2. Despliega una ventana donde se permite modificar qué piezas y/o componentes forman el mismo. |
| 3. Agrega o quita algun compuesto del componente y hace click en el botón Editar. |  |
|  | 4. Muestra un mensaje de confirmación de la modificación. |
| **Curso Alternativo 1** | |
| 1. Quita todos los compuestos del componente y hace click en el botón Editar. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |

# Eliminar un componente reusable

|  |  |
| --- | --- |
| **Descripción:** El usuario elimina un componente existente | |
| **Actores:** Usuario | |
| **Precondición:** El componente debe estar ingresado en el sistema | |
| **Postcondición:** Elimina un componente | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de componentes, selecciona uno y hace click en el botón Eliminar. |  |
|  | 1. Le pregunta al usuario si está seguro de querer eliminar el componente. |
| 1. Selecciona la opción afirmativa. |  |
|  | 1. Elimina el componente y muestra un mensaje de confirmación. Vuelve a la ventana de Administración de componentes. |
| **Curso Alternativo 1** | |
| 1. Selecciona la opción afirmativa. |  |
|  | 1. No elimina el componente ya que forma parte de algún componente o diseño por lo que muestra un mensaje de error |
| **Curso Alternativo 2** | |
| 1. Selecciona la opción negativa. |  |
|  | 1. Vuelve a la ventana de Administración de componentes. |

# Ver un componente reusable

|  |  |
| --- | --- |
| **Descripción:** El usuario visualiza un componente existente | |
| **Actores:** Usuario | |
| **Precondición:** El componente debe estar ingresado en el sistema | |
| **Postcondición:** Visualiza un componente | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de componentes, selecciona uno y hace click en el botón Ver. |  |
|  | 1. Abre una ventana con los datos del componente (marca y modelo) y sus compuestos (piezas y/o componentes). |
| 1. Selecciona el botón Volver. |  |
|  | 1. Vuelve a la ventana de Administración de componentes. |

# Agregar un diseño

|  |  |
| --- | --- |
| **Descripción:** El usuario agrega un diseño | |
| **Actores:** Usuario | |
| **Precondición:** Que hayan piezas ingresadas previamente | |
| **Postcondición:** Se agrega un diseño | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. Selecciona el ítem Drone en la pantalla principal. |  |
|  | 1. Despliega un submenú con los ítems Administar Diseños y Administrar Drones. |
| 1. Selecciona el ítem Administar Diseños. |  |
|  | 1. Muestra un listado de los diseños ingresados hasta el momento y opciones para agregar y modificar un diseño. |
| 1. Selecciona el botón para Agregar un diseño. |  |
|  | 1. Despliega una ventana para ingresar los datos del diseño y la selección de piezas y/o componentes que lo pueden conformar. |
| 1. Ingresa los datos del diseño, selecciona las piezas y/o componentes que lo forman y hace click en el botón Continuar. |  |
|  | 1. Registra el diseño, y muestra un mensaje de confirmación. Abre una ventana donde se deben configurar las alarmas de los sensores del diseño. |
| 1. Ingresa los valores de la alarma de cada sensor haciendo click en Setear para cada uno. Hace clik en Terminar. |  |
|  | 1. Muestra un mensaje de confirmación del ingreso de las alarmas y vuelve a la ventana del Administración de Diseños. |
| **Curso Alternativo 1** | |
| 1. Ingresa los datos sin seleccionar ninguna pieza o componente para el diseño y hace click en Continuar. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |
| **Curso Alternativo 2** | |
| 1. Ingresa los datos, donde el nombre del diseño ya existe en el sistema. |  |
|  | 1. Muestra un mensaje de error indicando que ya existe un diseño con ese nombre. |
| **Curso Alternativo 3** | |
| 1. Ingresa los datos pero no selecciona ningun GPS ni Velocímetroy hace click en Continuar. |  |
|  | 1. Muestra un mensaje de error indicando que el diseño debe contener al menos un GPS y un Velocímetro. |
| **Curso Alternativo 4** | |
| 1. Ingresa los valores de la alarma de cada sensor haciendo click en Setear para cada uno. |  |
|  | 1. Muestra un mensaje de error indicando que el valor de la alarma debe estar dentro del rango del mínimo y máximo del sensor. |
| **Curso Alternativo 5** | |
| 1. Hace click en el botón **Cancelar.** |  |
|  | 1. El sistema volverá a la ventana de Administración de diseños. |

# Editar un diseño

|  |  |
| --- | --- |
| **Descripción:** El usuario edita un componente existente | |
| **Actores:** Usuario | |
| **Precondición:** El componente debe estar ingresado en el sistema | |
| **Postcondición:** Modifica un componente | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de diseños, selecciona un diseño y hace click en el botón Editar. |  |
|  | 2. Despliega una ventana donde se permite modificar qué piezas y/o componentes forman el mismo. |
| 3. Agrega o quita algun compuesto del diseño y hace click en el botón Editar. |  |
|  | 4. Muestra un mensaje de confirmación de la modificación, guardando el diseño con el mismo nombre pero con un número más de versión, y abre una ventana donde se deben configurar las alarmas de los sensores del diseño. |
| 1. Ingresa los valores de la alarma de cada sensor haciendo click en Setear para cada uno. Hace clik en Terminar. |  |
|  | 1. Muestra un mensaje de confirmación del ingreso de las alarmas y vuelve a la ventana del Administración de Diseños. |
| **Curso Alternativo 1** | |
| 1. Quita todos los compuestos del diseño y hace click en el botón Editar. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |
| **Curso Alternativo 2** | |
| 1. Ingresa los datos pero no selecciona ningun GPS ni Velocímetroy hace click en Continuar. |  |
|  | 1. Muestra un mensaje de error indicando que el diseño debe contener al menos un GPS y un Velocímetro. |
| **Curso Alternativo 4** | |
| 1. Ingresa los valores de la alarma de cada sensor haciendo click en Setear para cada uno. |  |
|  | 1. Muestra un mensaje de error indicando que el valor de la alarma debe estar dentro del rango del mínimo y máximo del sensor. |
| **Curso Alternativo 5** | |
| 1. Hace click en el botón **Cancelar.** |  |
|  | 1. El sistema volverá a la ventana de Administración de diseños. |

# Agregar un dron

|  |  |
| --- | --- |
| **Descripción:** El usuario agrega un dron | |
| **Actores:** Usuario | |
| **Precondición:** Que haya algún diseño ingresadas previamente | |
| **Postcondición:** Se agrega un dron | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. Selecciona el ítem Drone en la pantalla principal. |  |
|  | 1. Despliega un submenú con los ítems Administar Diseños y Administrar Drones. |
| 1. Selecciona el ítem Administar Drones. |  |
|  | 1. Muestra un listado de los drones ingresados hasta el momento y opciones para agregar, modificar y eliminar un dron. |
| 1. Selecciona el botón para Agregar un dron. |  |
|  | 1. Despliega una ventana para ingresar los datos del dron, incluyendo la selección del diseño. |
| 1. Ingresa los datos del dron y selecciona el diseño que tiene y hace click en Agregar. |  |
|  | 1. Registra el dron, y muestra un mensaje de confirmación. Vuelve a la ventana de Administación de drones. |
| **Curso Alternativo 1** | |
| 1. Ingresa los datos dejando un campo vacío. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |

# Editar un dron

|  |  |
| --- | --- |
| **Descripción:** El usuario edita un dron existente | |
| **Actores:** Usuario | |
| **Precondición:** El dron debe estar ingresado en el sistema | |
| **Postcondición:** Modifica un ron | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de drones, selecciona un dron y hace click en el botón Editar. |  |
|  | 2. Despliega una ventana donde se permite modificar los datos del dron. |
| 3. Modifica algún dato y hace click en Editar. |  |
|  | 4. Muestra un mensaje de confirmación de la modificación. |
| **Curso Alternativo 1** | |
| 1. Ingresa los datos dejando un campo vacío. |  |
|  | 1. Muestra un mensaje de error indicando que faltan ingresar datos. |

# Eliminar un dron

|  |  |
| --- | --- |
| **Descripción:** El usuario elimina un dron existente | |
| **Actores:** Usuario | |
| **Precondición:** El dron debe estar ingresado en el sistema | |
| **Postcondición:** Elimina un ron | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. En a ventana de Administración de drones, selecciona un dron y hace click en el botón Eliminar. |  |
|  | 1. Pregunta si está seguro de querer eliminar el dron seleccionado. |
| 1. Selecciona la opción afirmativa |  |
|  | 1. Se elimina el dron y muestra un mensaje de confirmación de la eliminación. |
| **Curso Alternativo 1** | |
| 1. Selecciona la opción negativa. |  |
|  | 1. Vuelve a la ventana de Administración de drones. |

# Inicar una simulación

|  |  |
| --- | --- |
| **Descripción:** El usuario inicia una simulación | |
| **Actores:** Usuario | |
| **Precondición:** Que haya algún dron ingresado previamente | |
| **Postcondición:** Se inicia una simulación | |
| **Requerimientos no funcionales:** 1, 3, 5 y 6 | |
| **Curso Normal** | |
| **Usuario** | **Sistema** |
| 1. Selecciona el ítem Simulación en la pantalla principal. |  |
|  | 1. Despliega un submenú con el ítem Iniciar Simulación. |
| 1. Selecciona Inicar Simulación. |  |
|  | 1. Muestra un listado de los drones ingresados hasta el momento. |
| 1. Selecciona un dron. |  |
|  | 1. Muestra los valores inicales del GPS y el Velocímetro permitiendo su reingreso y además solicita el ingreso de las coordenadas finales de vuelo. |
| 1. Ingresa los datos requeridos y hace click en Continuar. |  |
|  | 1. Abre una ventana donde se puede modificar la configuración de las alarmas de los sensores del diseño. |
| 1. Setea las alarmas y hace click en Continuar. |  |
|  | 1. Abre la ventana de la Consola Administrativa donde de ven los valores de los sensores y se permite modificarlos. |
| 1. Hace click en el botón Ver Panel. |  |
|  | 1. Abre la Consola del Piloto, donde se ven los valores de los sensores, permitiendo al usuario visualizar ambas ventanas de las consolas a la vez. |
| 1. En la Consola Administrativa, modifica el valor de algún sensor y hace click en Modificar. |  |
|  | 1. Refleja los cambios en ambas ventanas, indicando en la consola del Piloto si se disparó alguna alarma. A su vez, cada 30 segundos, va guardando los datos de la simulación. |
| 1. Repite el paso 13 las veces deseadas hasta indicar en el GPS las coordenadas finales de vuelo. |  |
|  | 1. Indica que el dron ha llegado al fin de su recorrido y muestra una ventana con el Reporte del vuelo. |
| 1. Hace click en Aceptar. |  |
|  | 1. Vuelve al Menú Principal. |
| **Curso Alternativo 1** | |
| 1. Ingresa los datos requeridos sobre las coordenadas finales de vuelo y hace click en Continuar. |  |
|  | 1. Muestra un mensaje de error indicando que los valores exceden los límites mínimo y máximo permitidos. |
| **Curso Alternativo 2** | |
| 1. Repite el paso 13 las veces deseadas hasta que un sensor se encuentre en alarma urgente. |  |
|  | 1. Muestra una imagen que indica el estado de alarma, y al paso de 5 segundos termina la simulación, destruyendo el dron e indicando que ha sido destruido a causa de que la alarma excedió los 5 segundos. Muestra una ventana con el Reporte del vuelo. |

# Glosario:

* **Datos de una pieza**: los datos que tiene cada pieza son la marca, el modelo, la presición, el mínimo y máximo del valor de su sensor y su tipo (Termómetro, Barómetro, Velocímetro, GPS, Altímetro y Acelerómetro).
* **Ventana de Administración de Piezas**: es la ventana donde se listan las piezas ingresadas en el sistema, con opciones de Agregar, Modificar y Eliminar una pieza. Se accede desde el Menu Principal, haciendo click en Componentes de vuelo y luego en Administrar Piezas.
* **Ventana de Administración de Componentes**: es la ventana donde se listan los componentes ingresadas en el sistema, con opciones de Ver, Agregar, Modificar y Eliminar un componente. Se accede desde el Menu Principal, haciendo click en Componentes de vuelo y luego en Administrar Componentes Reusables.
* **Ventana de Administración de Diseños**: es la ventana donde se listan los diseños ingresadas en el sistema, con opciones de Agregar y Modificar un diseño. Se accede desde el Menu Principal, haciendo click en Dron y luego en Administrar Diseños.
* **Ventana de Administración de Drones**: es la ventana donde se listan los drones ingresadas en el sistema, con opciones de Agregar, Modificar y Eliminar un dron. Se accede desde el Menu Principal, haciendo click en Dron y luego en Administrar Drones.