1.Introdución

Este documento farase un recorrido por toda a fase de desenvolvemento do proxecto, aportando os diagramas de deseño do sistema, co fin de facilitar as labores de mantenemento e ampliación do sistema nun futuro.

A metodoloxía de desenvolvemento que se utilizou foi unha metodoloxía basada en prototipos. Esta metodoloxía básase na construcción de prototipos, isto é, a construcción dunha aplicación que conten un subconxunto dos obxetivos e funcionalidades do sistema. Mediante a construcción dun prototipo evolutivo estímase que coa finalización do terceiro prototipo éste sexa o sistema final completo.

Nesta metodoloxia primeiramente faise una análisis preliminar dos obxetivos do sistema e determínase a grandes rasgos a estructura que terá. Logo pasarase a creación dos prototipos para finalmente entrar na fase de desenvolvemento do producto final que consiste nada mais ca convertir o último prototipo desenvolvido no sistema final.

A creación de cada prototipo dividese en tres fases. Unha fase de deseño rápido onde se observará o deseño da estructura do sistema a través dun diagrama de clases, e o conxunto de funcionalidades mediante un diagrama de casos de uso. Ademais para comprender mellor o funcionamento interno dos casos de uso aportaranse diagramas de secuencia de sistema. A seguinte fase é a fase de construcción onde se implementará o prototipo seguindo o deseño aportado e finalemente se faran unhas pequenas probas de unidade. E por último a dase de evaluación e retroalimentación na cal nos reuniremos có tutor do proxecto para facer unha demostración de funcionamento do sistema e determinar os erros que se deberán correxir no seguinte prototipo e as novas funcionalidades que debe ter.

A utilización desta metodoloxía de desenvolvemento permitenos crear rápidamente o sistema e conseguir un sistema moi probado debido a que se publicaron diversos prototipos e cada un deles foi sometido a unha pequena fase de proba real. Pero como contrapartida o nivel de documentación é mínimo, redúcese únicamente ós diagramas utilizados para o deseño rápido de cada un dos prototipos.

2. Investigación preliminar

O proxecto que vaise a desenvolver trátase dun sinxelo sistema que permita a creación de interfaces gráficas ca API Winforms de C# para mais tarde poder importalas nun proxecto en desenvolvemento axeno a este sistema.

O sistema comporase da ferramenta de deseño e dunha librería para poder utilizar as uinterfaces creadas, noutro proxecto en desenvolvemento.

A ferramenta debe permitir, non todos os controles existentes na API, pero si os máis comuns e utilizados. Da mesma forma o nivel de personalización destes controles debe centrarse nas propiedades mais importantes.

Tras analizar os distintos controles da API Winforms púidose xerar un diagrama de clases (Figura 7) moi sinxelo de modo que nos poida servir para ter unha idea inicial de como será o sistema, e así, a partir del deseñar a estructura dos seguintes prototipos en base ás funcionalidades e requisitos que cubrirán.

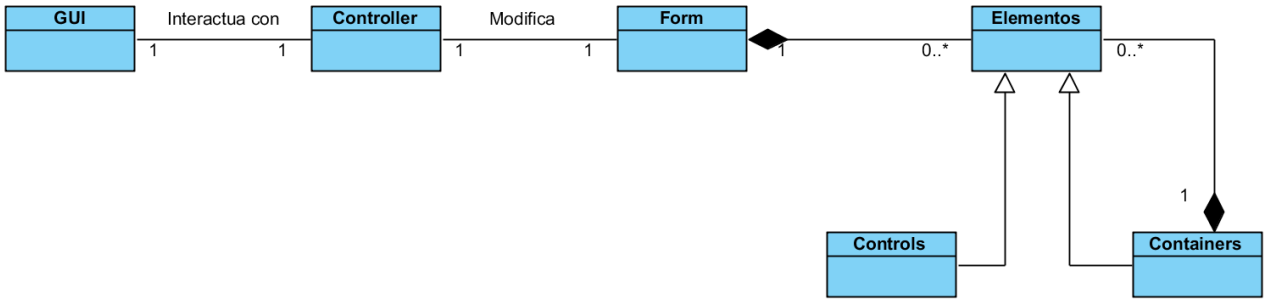

Figura 7. Diagrama de clases

GUI: Interfaz gráfica da ferramenta.
Controller: Clase intermediaria entre a interfaz e o formulario.
Form: Clase que representa o formulario que se está a deseñar coa ferramenta.
Elementos: Representa ó conxunto de obxetos que se lle poden añadir á interfaz.
Controls: Correspóndese ós elementos simples da API Winforms.
Containers: Elementos da API Winforms que serven para conter outros elementos.

Con respecto ó deseño da librería, o diagrama de clases correspondente é o mesmo ó da ferramenta pero eliminando a clase "GUI".

Para poder comprender mellor as funcionalidades principais que se esperan da ferramenta de deseño, e nos seguintes prototipos poder repartilas entre os prototipos planificados, onde se extenderán mais, creouse un pequeno diagrama de casos de uso (Figura 8).

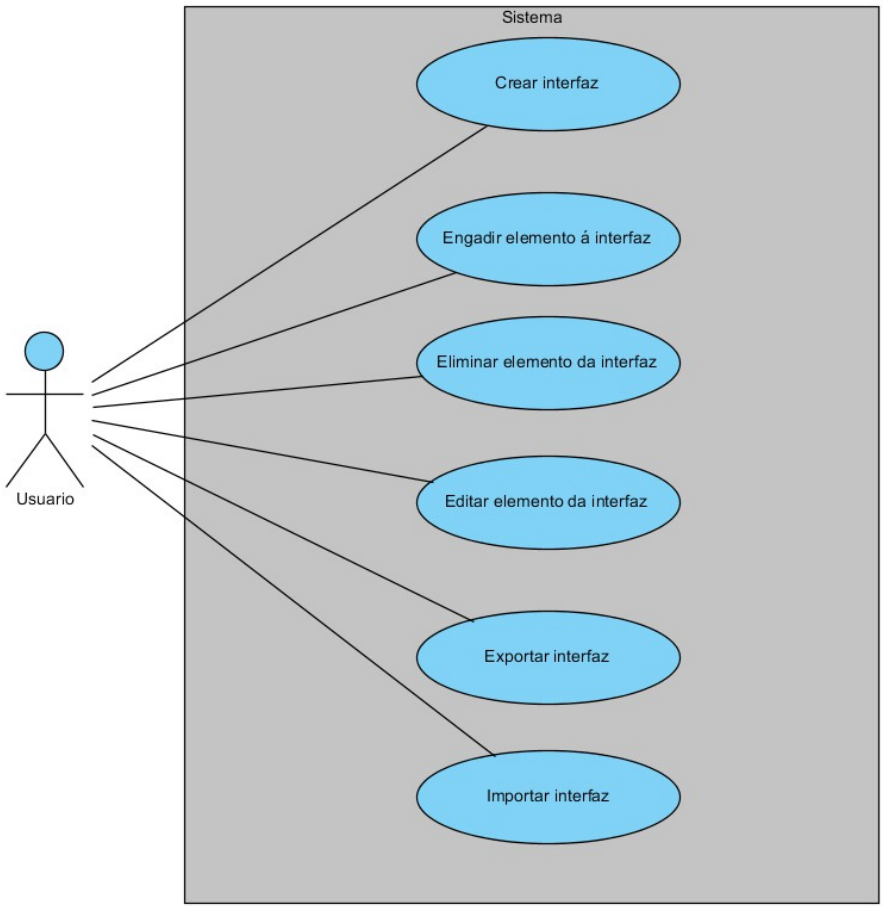

Figura 8. Diagrama de casos de uso

Crear Interfaz

Este caso de uso indica que o sistema deber permitir a creación dunha nova interfaz gráfica para poder deseñala na ferramenta.

Engadir elemento á interfaz

O sistema debe permitir engadir múltiples elementos á interfaz en deseño.

Editar elemento

Debe ser posible editar as propiedades correspondentes ós elementos que compoñen a interfaz en deseño.

Eliminar elemento da interfaz

Debe existir a posibilidade de eliminar elementos presentes na interfaz que se está a deseñar.

Exportar interfaz

Esta funcionalidade do sistema fai referencia a que se debe permitir exportar a interfaz que se ha deseñado coa ferramenta para asi dispoñer dela noutro proxecto.

Importar interfaz

Debe ser posible importar a interfaz que anteriormente foi exportada para poder continuar co seu deseño o modificala.

No caso da biblioteca, tan só existe un caso de uso, este é "Importar interfaz". Da mesma forma que se importa a interfaz para podela modificar, tamen se pode importar nun proxecto C# en desenvolvemento.

Para ter unha idea un pouco mais concreata de cal será o funcionamento interno do sistema para cada un dos casos de uso específicados, desenvolvéronse os correspondentes diagramas de secuencia os cales pódense apreciar nas Figuras 9,10,11,12,13 e 14.

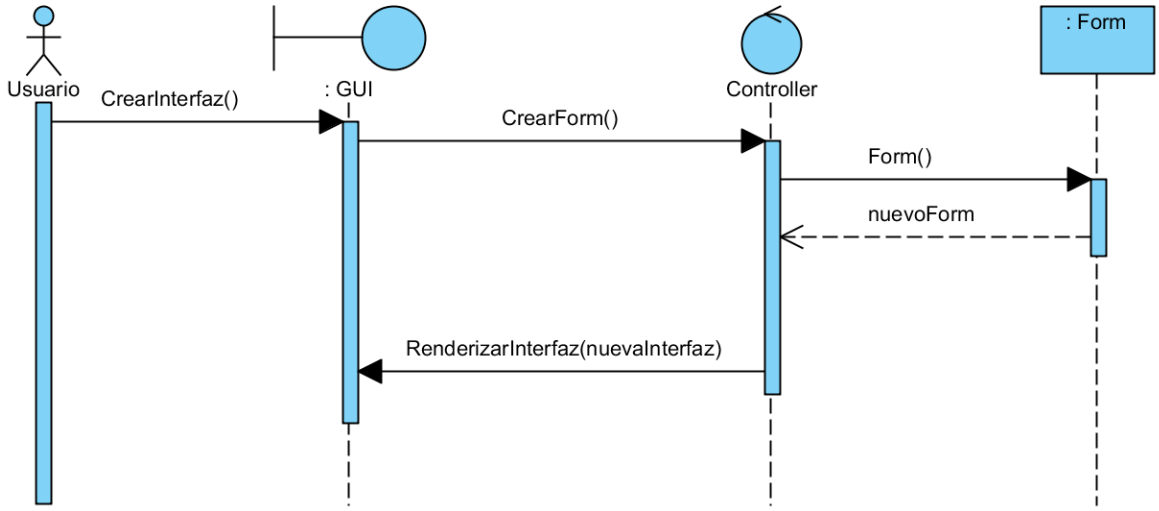

Figura 9. Diagrama de secuencia. Crear interfaz.

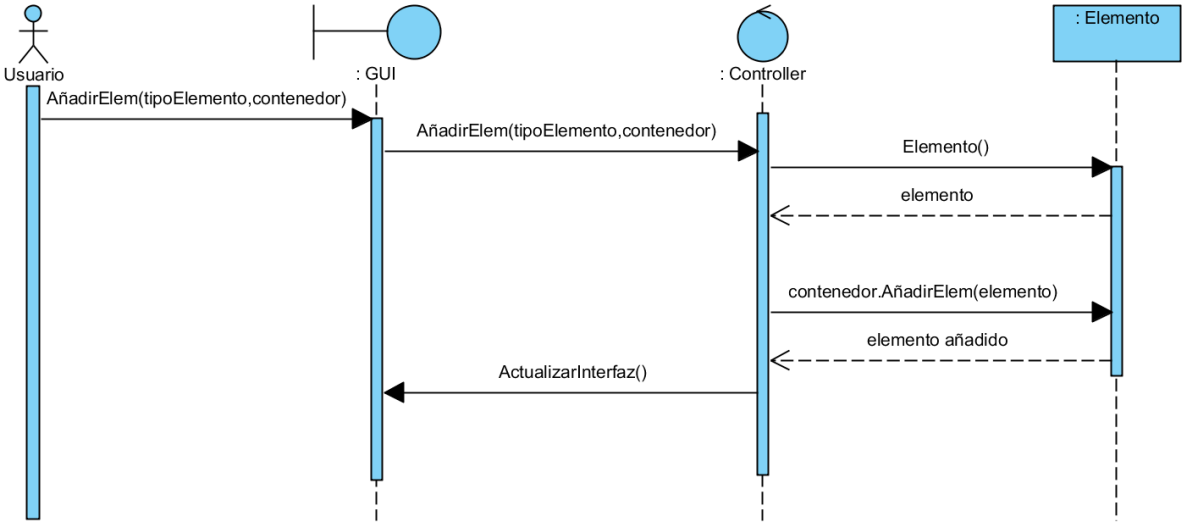

Figura 10. Diagrama de secuencia. Emgadir elemento.

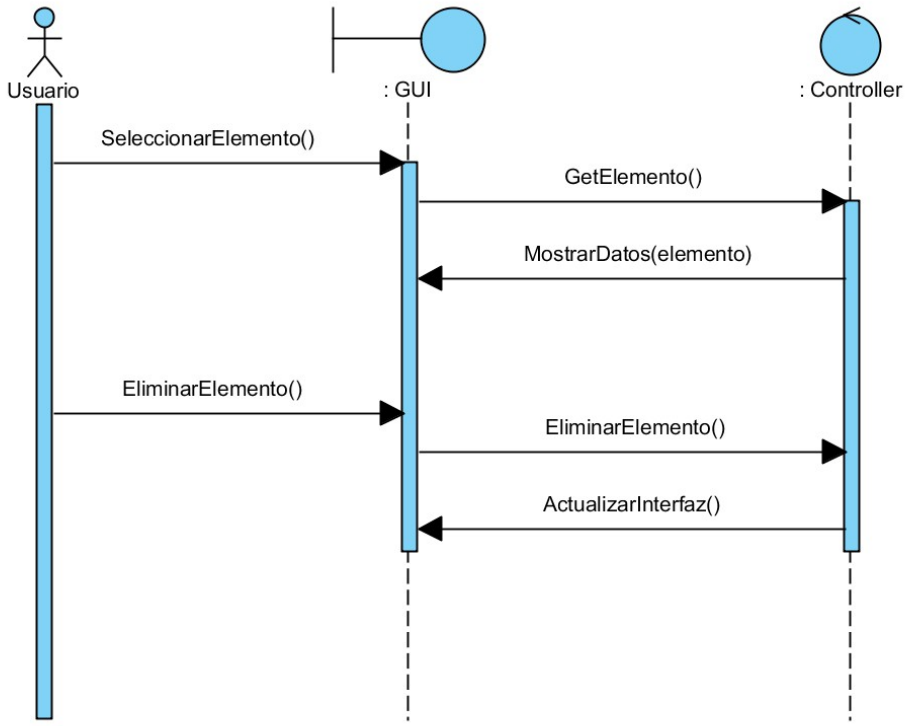

Figura 11. Diagrama de secuencia. Eliminar elemento.

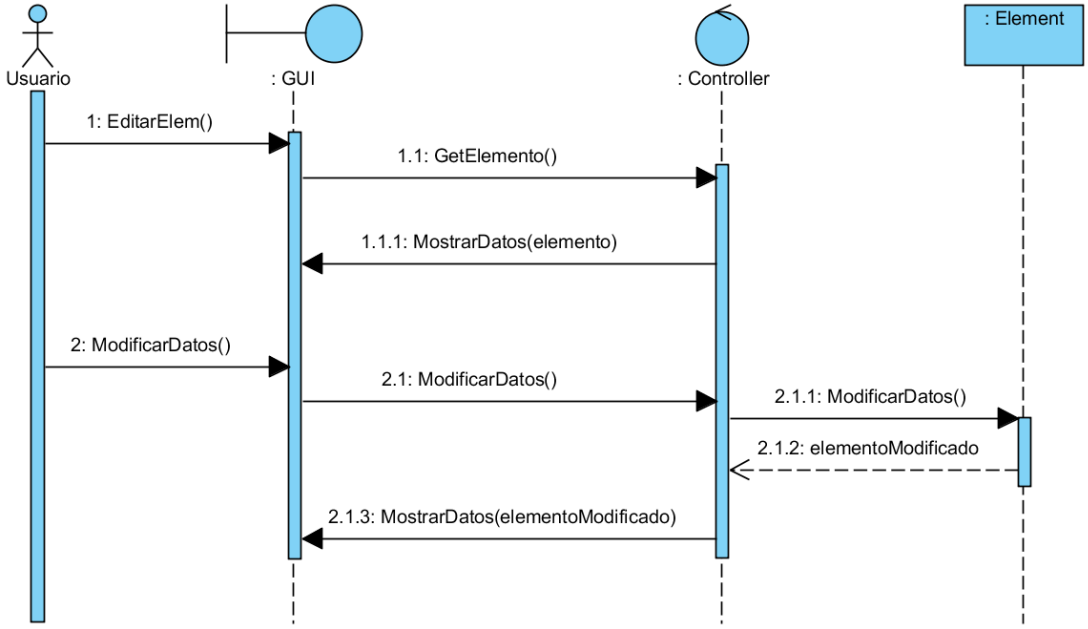

Figura 12. Diagrama de secuencia. Editar elemento.

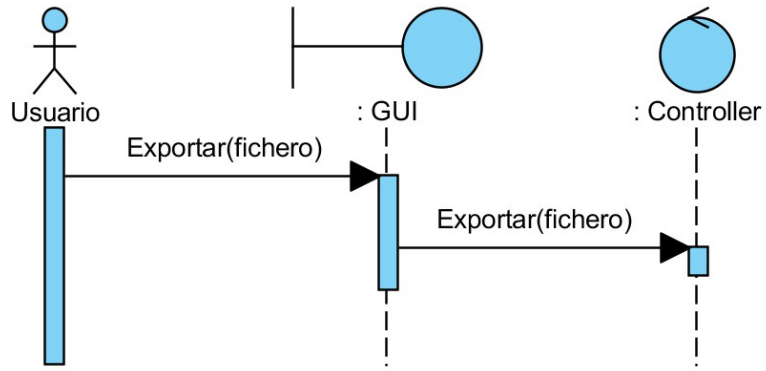

Figura 13. Diagrama de secuencia. Exportar interfaz.

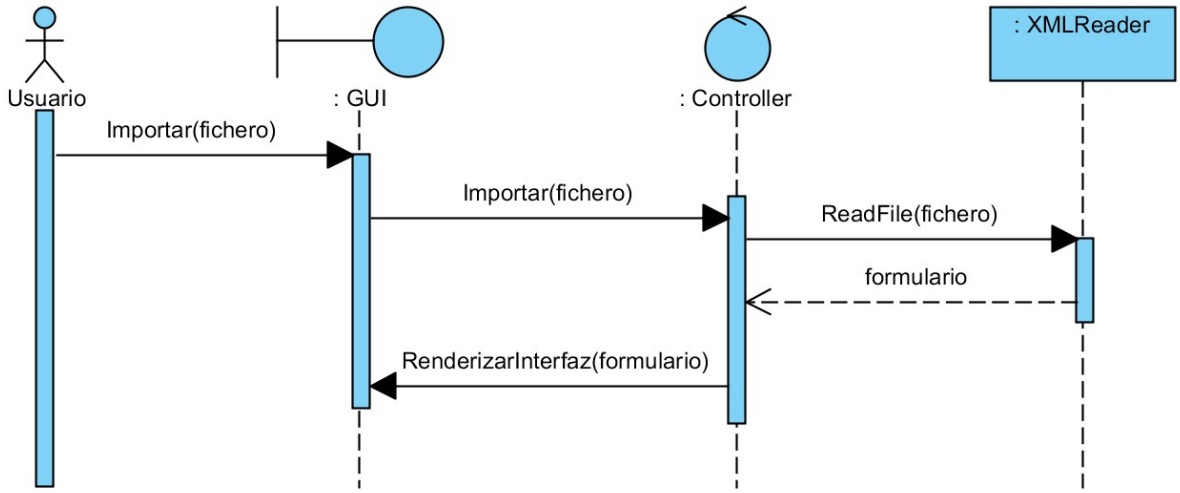

Figura 14. Diagrama de secuencia. Importar interfaz.

Tras esta primera aproximación a la arquitectura y el funcionamiento del sistema ya podemos comenzar con el desarrollo de los prototipos.

3. Primer prototipo

3.1. Deseño rápido

Para o primeiro prototipo tívose como obxetivo a creación dunha ferramenta de deseño sinxela. Esta ferramenta dará soporte á creación da interfaz e poderlle engadir uns poucos elementos. Estes elementos son:

- Containers
  - Border
  - HBox
  - VBox
  - Grid
- Controles
  - Button
  - Label
  - TextBox

Éste é o conxunto de controles que se pensou que era o mais sinxelo deixando os mais complicados para os seguintes prototipos xa que neste primeiro prototipo tense que construir a interfaz gráfica completa.

Para deseñar a interfaz, fíxose o boceto da Figura 15, no cal pódese observar que a interfaz se compoñerá dun treeview no lateral esquerdo, no cal se mostrarán os diferentes elementos engadidos na interfaz e desenvolvemento. Na parte dereita pódese ver unha tabla na cal se poderan visualizar i editar as propiedades dun dos elementos engadidos á interfaz. Finalmente cóntase cunha zona central, a zona de traballo, onde se poderá ir visualizando a interfaz que se está a deseñar.

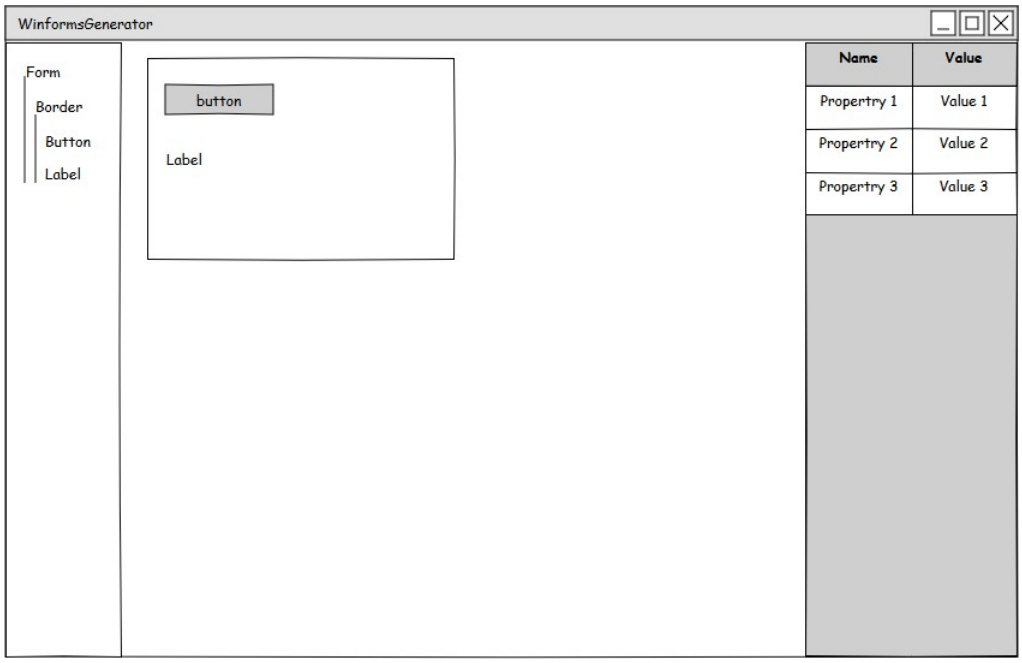

Figura 16. Boceto da interfaz de usuario

Tendo en conta estas funcionalidades principales coas que contará o primero prototipo, na Figura 17 pódese observar o diagrama de casos de uso equivalente.

Figura 17. Diagrama de casos de uso do Primeiro Prototipo

Crear interfaz

Cada vez que se inicia a ferramenta de deseño de interfaces, créase unha nova interfaz en blanco.

Engadir elemento á interfaz

Para executar esta acción débese executar dende a interfaz a accion de engadir nun elemento concreto da interfaz e seleccionar o elemento que se quere engadir.

Tras esto créase un novo elemento do tipo indicado e engádese ó elemento seleccionado, un *Container* ou o propio *Form*.

Copiar elementos

Primeiramente débese seleccionar o elemento a copiar e logo executar a accion de copiar. Mediante esta acción cópiase tanto o elemento coma os elementos que este contén se se trata dun *Container.*

Pegar Elemento

Tras realizar unha acción de copiado habilítase a posibilidade de pegar o elemento gardado. Tras seleccionar o elemento, un *Container* ou o *Form*, no cal se quere pegar, engádese a ese contenedor o elemento e subelementos que se copiaran.

Eliminar Elemento

Primeiramente débese seleccionar un elemento da interfaz que queremos eliminar. Tras esto seleccionase a acción de elimnar elemento e este desaparece da interfaz en desenvolvemento

Cortar Elemento

Simplemente trátase de executar a accion de copiado e a de borrado. Debese seleccionar o elemento que queremos cortar e este cópiase e se elimina da interfaz en desenvolvemento.

Tendo en conta as funcionalidades indicadas no diagrama, o boceto da interfaz e o diagrama de clases da investigación preliminar, desenvolveuse o diagrama de clases mais detallado da Figura 18.

Debuxar Interfaz

Tras realizar calquera de estas acción anteriores, excepto o copiado, actualízase toda a interfaz da aplicación. Esta actualización consiste en redibuxar o panel esquerdo que conten o *TreeView* dos

elementos que compoñen a interfaz que se esta a deseñar, e en redibuxar a vista previa desa interfaz no panel de traballo central.
Apartir de esta definicion de casos de uso desenvolveuse os diagramas de secuecia de sistema que se aportan a continuación. Neles indicase dunha forma un pouco abreviada o funcionamento interno real do sistema para cada unha das funcionalidades.

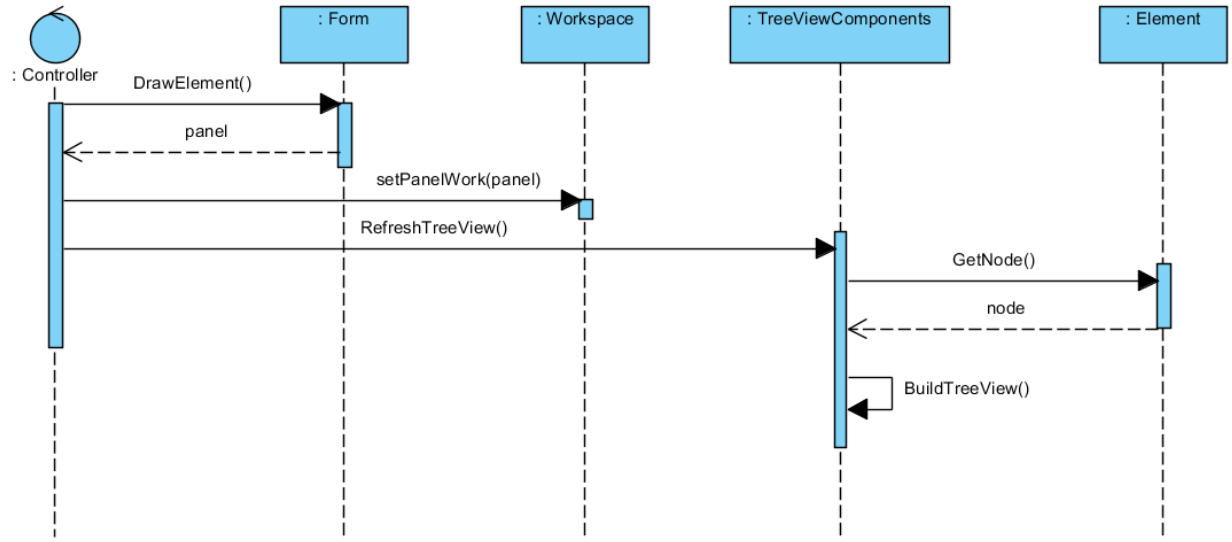

Figura 18. Diagrama de secuencia. Debuxar interfaz

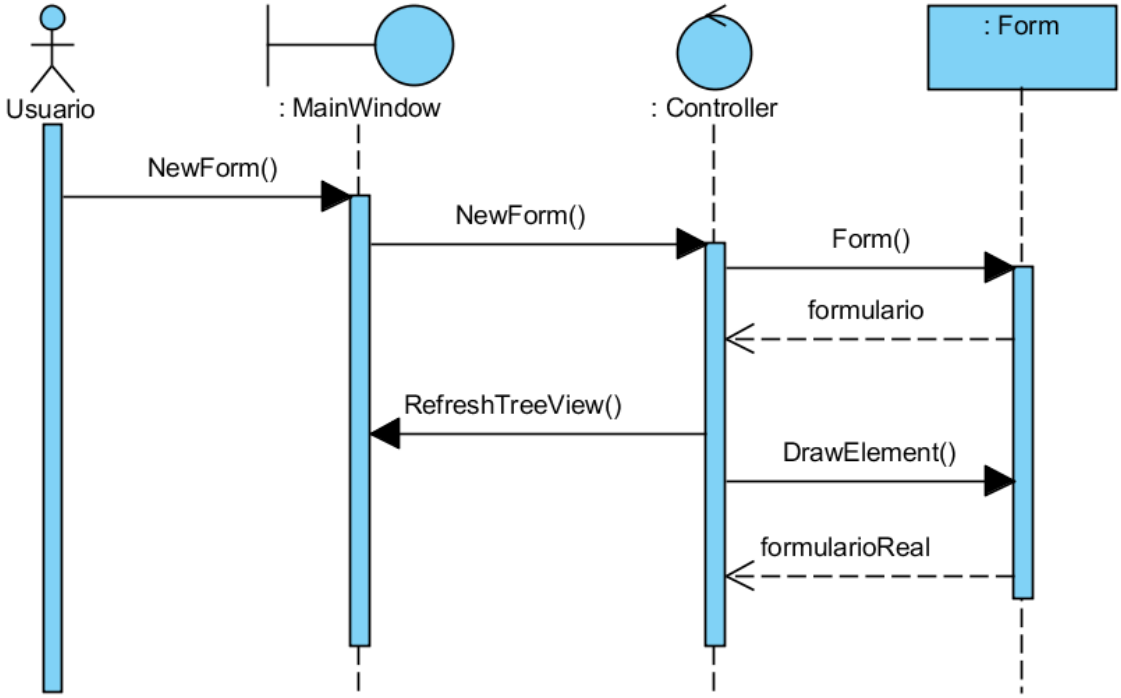

Figura 19. Diagrama de secuencia. Crear interfaz

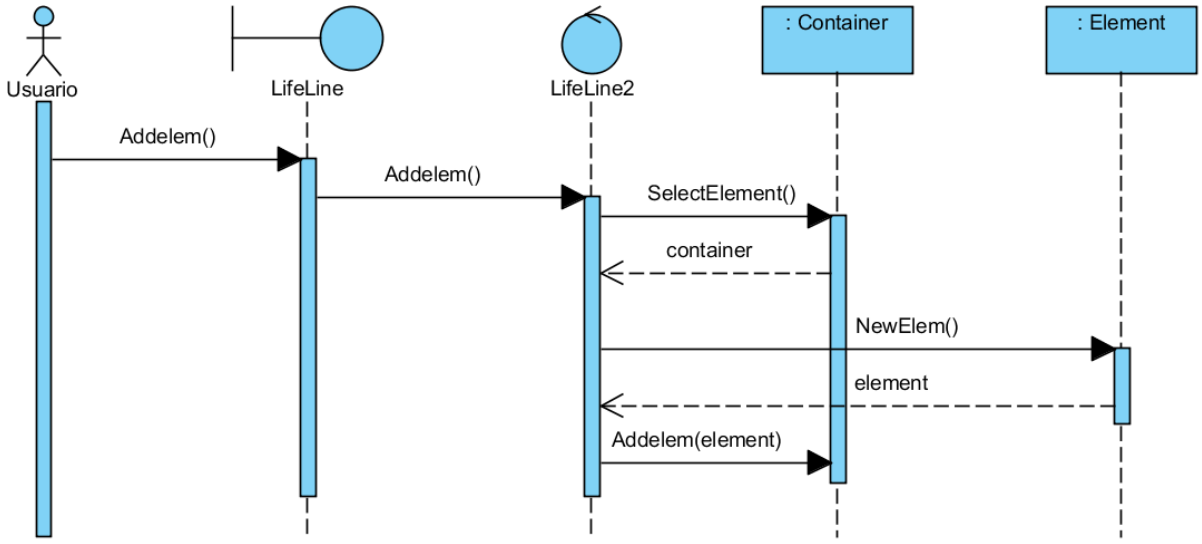

Figura 20. Diagrama de secuencia. Engadir elemento

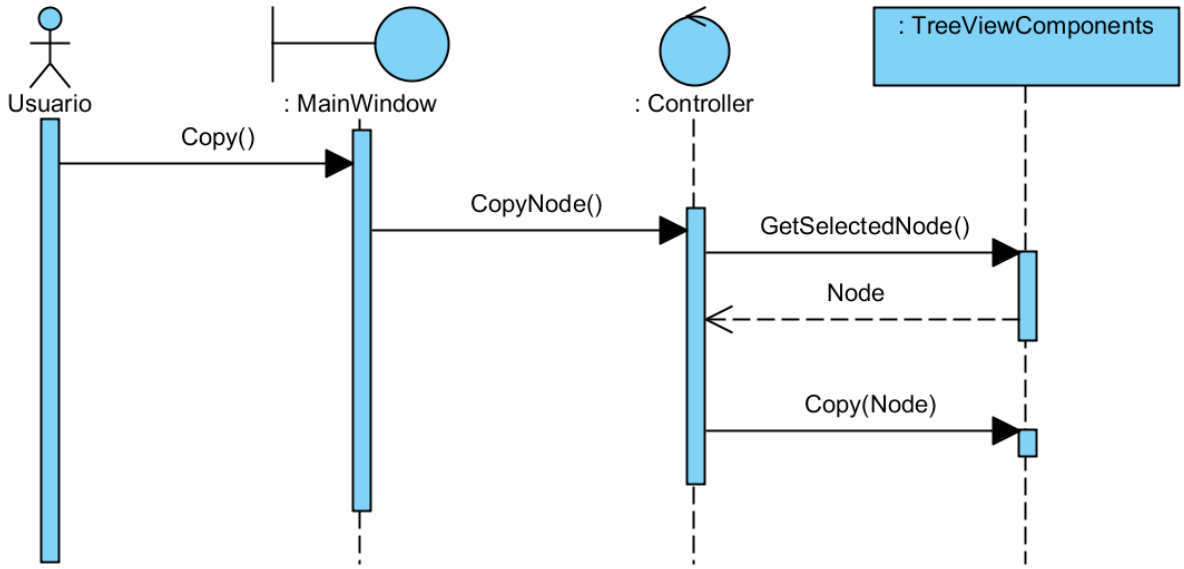

Figura 21. Diagrama de secuencia. Copiar elemento

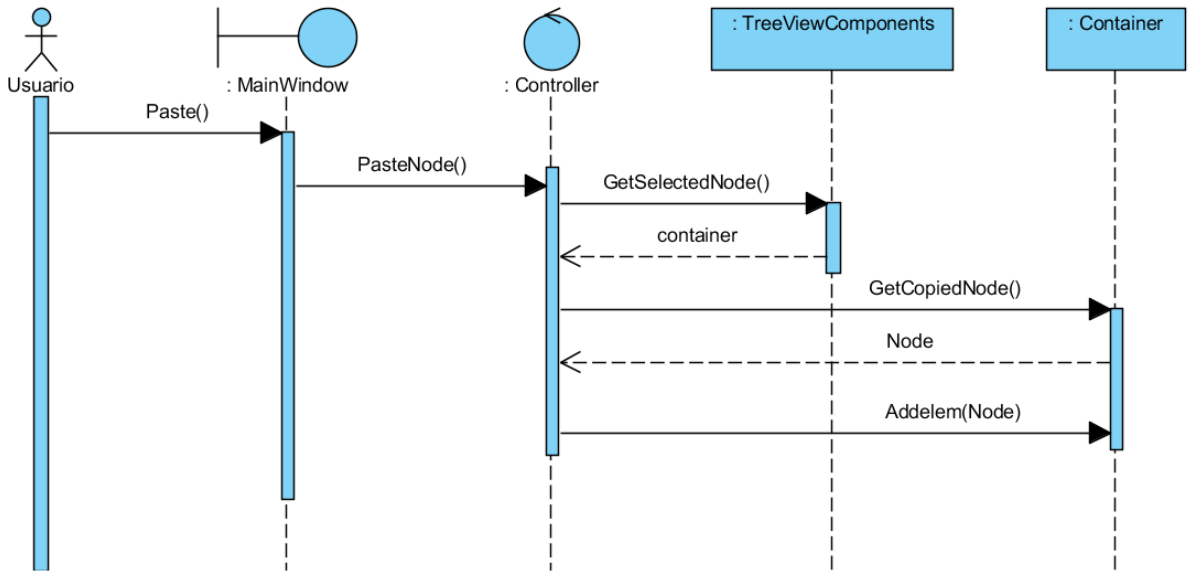

Figura 22. Diagrama de secuencia. Pegar elemento

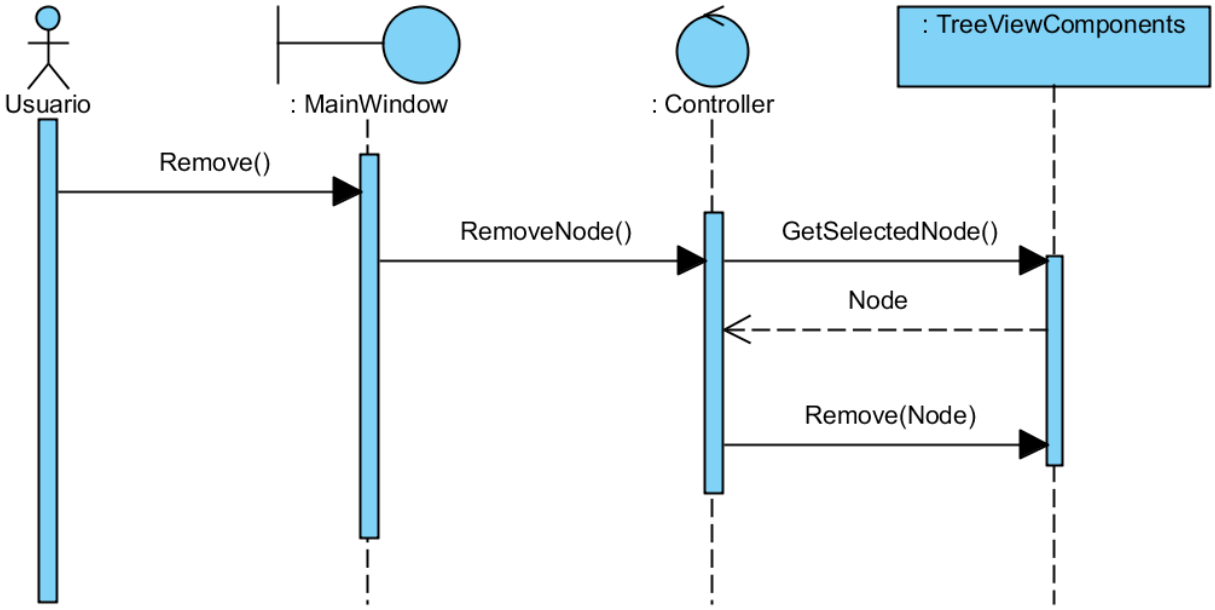

Figura 23. Diagrama de secuencia. Eliminar Elemento

Tendo en conta a especificación do funcionamento do sistema mediante os diagramas de secuencia, xerouse o diagrama de clases da Figura 24, no cal se detalla moito mais a arquitectura do sistema aportando atributos e funcións que deben tener cada unha das clases.

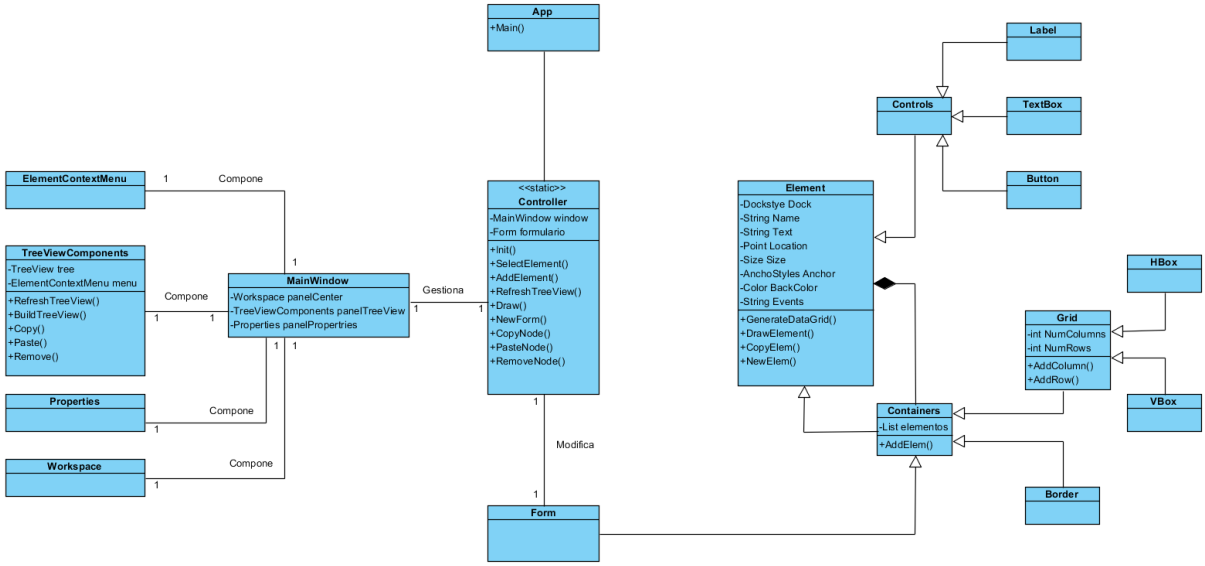

Figura 24. Diagrama de clases do Primeiro Prototipo

App

Esta clase trátase da clase principal do sistema onde se instancia toda a aplicación.

TreeViewComponents

Esta clase trátase dun panel da API Winforms que contén un TreeView, no cal se poderá observar a estructura da interfaz que se está a desenvolver coa ferramenta

- **Atributos:**
  - *Treeview tree:* TreeView que esquematiza os elementos que forman parte da interfaz en desenvolvemento
  - *ElementContextMenu menu*: ContextMenuStrip o cal se accede co click dereito do ratón e que contén as accions que se poder realizar sobre os elementos da interfaz en desenvolvemento.
- **Métodos:**
  - *BuildTreeView*: Este métod server para debuxar o TreeView a partir da lista de elementos que compoñen a interfaz.
  - *RefreshTreeView* : Método que serve para actualizar o TreeView e se lle chama tras realizar algunha acción que modifique a estructura da interfaz, accións tales como engadir novos elementos ou borralos.
  - *Copy*: Este método permite copiar unha seccion do TreeView para pegala mais tarde noutra parte e facendo que se recoloquen na interfaz os elementos representados por esa sección de TreeView.
  - *Paste*: Método para pegar unha sección de TreeView copiada previamente.
  - *Remove:* Con este método faise posible eliminar seccións do TreeView facendo que os elementos representados por esa sección tamen se eliminen da interfaz en desenvolvemento.

ElementContextMenu

Esta clase xera un ContextMenuStrip que conterá as accións que se poderán executar no sistema, tales como engadir novo elemento, copiar, cortar, pegar e borrar elementos, etc.

Properties

Representa o panel dereito da interfaz, onde existe un DataGridView que conterá as propiedades que se poderan modificar, dun elemento da interfaz seleccionado previamente.

Workspace

Esta clase conten o panel central da aplicación onde estará a visualización da interfaz que se esta a crear coa ferramenta.

MainWindow
Trátase da ventana principal e única da ferramenta. Está formada por as tres seccions que se puideron apreciar na Figura 16.

- **Atributos:**
  - *Workspace panelCenter*: Panel central da ferramenta onde se poderá observar a construcción da interfaz en desenvolvemento.
  - *TreeViewComponents panelTreeView*: Panel esquerdo da interfaz onde se poderá atopar o TreeView que representa a estructura da interfaz en desenvolvemento.
  - *Propertries panelPropertries*: Panel dereito onde se mostrarán as propiedades a modificar dun elemento da interfaz.

Element
Esta clase representa ós elementos que se poderán engadir a unha interfaz en creación.

- **Atributos:**
  - Todas as propiedades desta clase correspondense coas mesmas propiedades presentes nos controles da API Winforms.
- **Métodos:**
  - *GenerateDataGrid*: Xera a tabla cas propiedades do elemento seleccionado.
  - *DrawElement*: Xera o control da API Winforms equivalente e cos valores das propiedades indicadas na tabla de propiedades do elemento.
  - *CopyElem*: Copia o elemento seleccionado.
  - *NewElem*: Crea un novo elemento.

Controls
Tipo de *Element* que se corresponde cos elementos mais sinxelos que se poderán incluir nunha interfaz. Del extenden **Label**, **TextBox** e **Button**.
Containers
Tipos de *Elements* que conteñen a outros obxetos da clase *Element*.

- **Atributos:**
  - *List<Elements> elementos*: trátase da lista de elementos que son contidos no propio obxeto.
- **Métodos:**
  - *AddElem*: Método que engade elementos ó *Container.*

Desta clase extenden os elementos **Border** e **Grid**.
Border
Esta clase representa o control *Panel* da API Winforms.
Grid
Esta clase representa ó TableLayoutPanel da API Winforms.

- **Atributos**:
  - *int NumColumns*: Número de columnas que ten o *Grid.*
  - *int NumRows*: Número de filas que ten o *Grid.*
- **Métodos:**
  - *AddColumns*: Engade unha columna ó *Grid*.
  - *AddRow*: Engade unha fila ó *Grid*.

Desta clase extenden os elementos **Hbox** e **Vbox** que son tipos especiais de *Grid* pero cunha soa fila ou unha soa columna respectivamente.
Form
Esta clase representa o formulario da interfaz que se vai a construir coa aplicación. Extende de *Container* xa que realmente se trata dese tipo de obxeto ainda que é un pouco especial xa que unicamente existe unha instancia del o mesmo tempo.
Controller
Esta clase trátase dunha clase intermediaria entre á aplicación, a interfaz gráfica e o formulario da interfaz en desenvolvemento.

- **Atributos**
  - *MainWindow window*: Instancia da interfaz gráfica da aplicación.
  - *Form formulario:* Instancia do formulario da interfaz en desenvolvemento.
- **Métodos**
  - *Init*: Método que inicializa ó controlador creando o formulario para desenvolver a interfaz e instanciando a interfaz gráfica da ferramenta.
  - *SelectElement*: Selecciona un elemento da interfaz en desenvolvemento xerando a tabla de propiedades para poder editala.
  - *AddElement*: Engade un elemento ó formulario e as conseguintes accións na interfaz da aplicación.
  - *RefreshTreeView*: Actualiza o *TreeView* da aplicación.
  - *Draw*: Debuxa a visualización da interfaz que se esta a desenvolver.
  - *NewForm*: Crea un novo formulario.
  - *CopyNode*: Copia unha seccion do *TreeView*
  - *PasteNode*: Pega unha seccion do *TreeView*, copiada previamente, nunha posición.
  - *RemoveNode*: Elimina unha seccion do *TreeView.*

Tras este sinxelo proceso de deseño da interfaz e da arquitectura e tras a definición das funcionalidades que terá este primeiro prototipo xa podemos avanzar a fase de construcción.

3.2. Construcción do prototipo

Nesta fase levouse a cabo a implementación do primeiro prototipo. Hay que destacar que durante esta fase, e tendo en conta que este non é o sistema final e que sufrira cambios e engadidos, o codigo creado pretende ter como característica ser facilmente ampliable, e dicir, implementouse da forma mais xenérica que se puido para así facilitar a implementación dos seguintes prototipos. Un exemplo que pon en evidencia esta característica do código é a utilización en diversas ocasiones a libreira de C# *Reflection* que fai posible a creación de métodos que cando se modifica a clase á cal pertencen éstes non teñen que ser modificados directamente senón que coa modificación da clase modifícase o funcionamento do método. Por exemplo, se engadimos un novo tipo de elemento deberíamos engadilo o menú onde están todos os elementos dispoñibles para crear a interfaz. Pero gracias a *Reflection* non é necesario engadilo xa q çue ó executar ó método éste detecta todas as clases que son elementos que se poden engadir a interfaz e crea o menú con eles.
Tras a construcción do prototipo desenvolvense unha serie de probas de unidade co fin de facilitar futuras ampliacións ou modificacións do prototipo. Estas probas, desenvoltas con NUnit consisten na proba dos métodos das clases que extenden de *Element* para asi comprobar que os Controles da API Winforms que se xerán coa ferramenta son como se desexa que sexan.

3.3. Evaluación e retroalimentación do prototipo

Nesta fase realizouse la reunión con el tutor a modo de cliente, para facer unha demostración do funcionamiento do prototipo e determinar qué aspectos do prototipo deben cambiarse ou ampliarse, así como determinar as funcionalidades que deberanse incluir no siguiente prototipo do sistema.
Como puntos de mellora do prototipo decidiuse incluir na interfaz un menú con botóns que representen as accions de crear novo formulario, copiar, cortar e pegar elemento.
Para ampliar o sistema no seguinte prototipo decidiuse engadir as opcións de gardado e carga da interfaz en desenvolvemento xa que ata agora non se podia facer.Tamen pensouse en dar opción a probar a interfaz que se está a crear. Para isto pode que sexa útil comezar xa coa biblioteca de importación da interfaz noutro proxecto.

4. Segundo prototipo

4.1. Deseño rápido

Neste segundo prototipo o deseño inicial será mais sinxelo xa que tan só consiste en engadir as novas funcionalidades ó sistema que xa temos ata agora. Por isto de agora en adelante tan só se explicarán as novas función ou clases engadidas no prototipo.
A partir da reunión de evaluación do anterior prototipo se obtiveron as novas funcionalidades que

debe ter o sistema neste segundo prototipo. Na Figura 25 pódese observar un pequeno diagrama de casos de uso correspondente ás funcionalidades da ferramenta de deseño. E na figura 26 o diagrama de casos de uso da bibliteca para importar a interfaz nun proxecto externo.

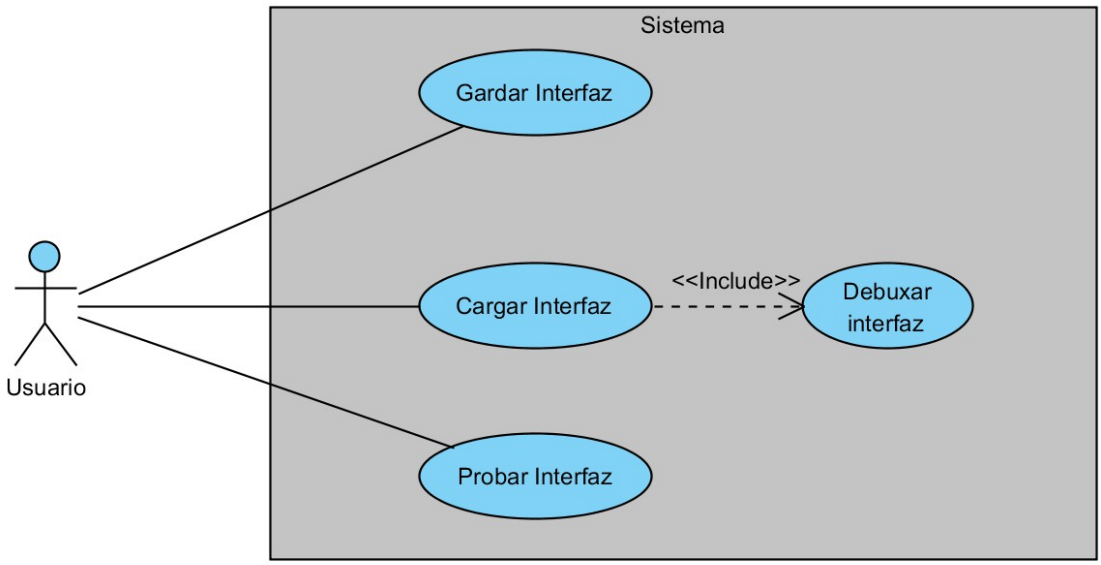

Figura 25. Diagrama de casos de uso da ferramenta de deseño no Segundo prototipo

Gardar interfaz

Mediante esta función se permitira gardar a interfaz deseñada na ferramenta nun ficheiro Xml para mais tarde poder utilizalo. Para facer o gardado da interfaz do ficheiro utilizarase a clase de C# XmlSerializer que permitirá serializar todos os elementos que compoñen a interfaz e gardalos cos valores das suas propiedades.

Cargar interfaz

Esta funcionalidade fará posible cargar na interfaz na nosa ferramenta para continuar co seu deseño. Para cargar o ficheiro coa información da interfaz tamen se utilizará a clase XmlSerilizer que permite deserializar os datos contidos no ficheiro Xml.

Probar Interfaz

Deberá ser posible facer una previsualización da interfaz deseñada coma se se estivera a utilizar nun proxecto externo, que é o obxetivo deste sistema. Para isto simplemente se creará unha ventana aparte que tan so conteña a interfaz tal e como foi deseñada.

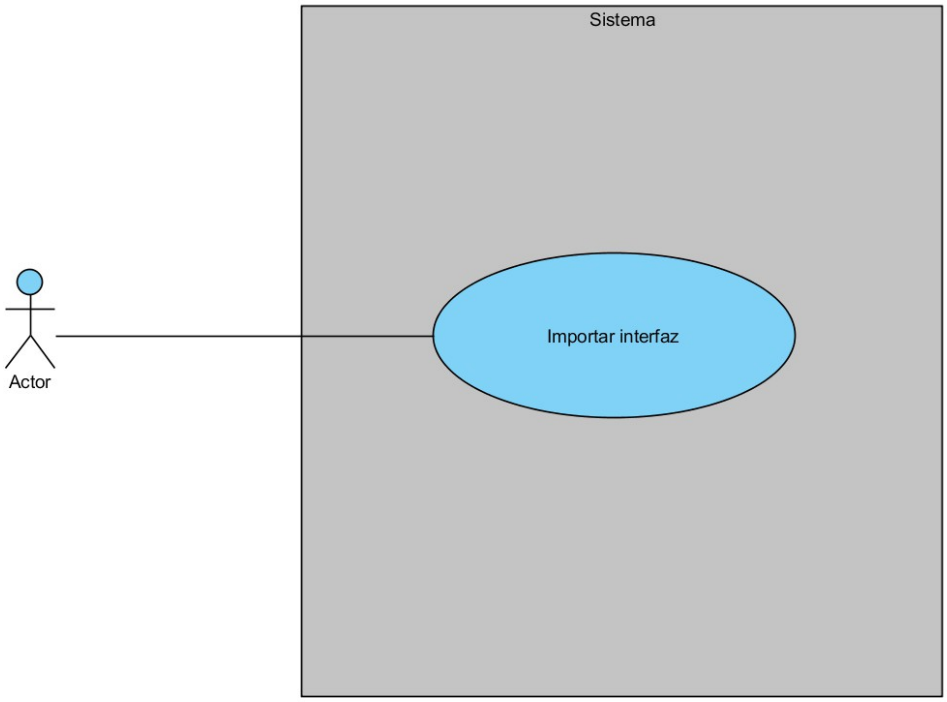

Figura 26. Diagrama de casos de uso da Biblioteca no Segundo prototipo

Importar interfaz

Esta funcionalidade fai referencia a que a biblioteca debe dar soporte á importación de interfaces dende un ficheriro Xml, nun proxecto C# calquera.

Para entender un pouco mais en detalle o funcionamento dos casos de uso tanto da ferramenta de deseño coma da biblioteca de importación deséñanse os diagramas de secuencia que se poden apreciar a continuación.

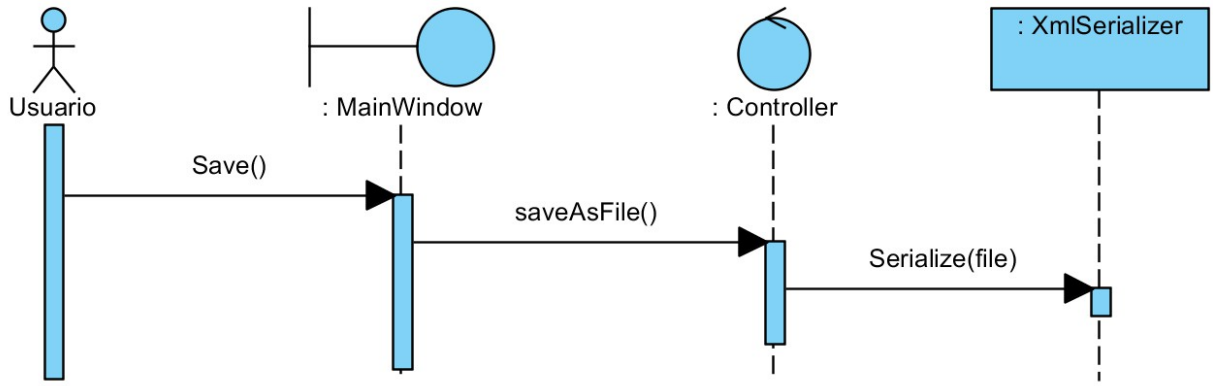

Figura 27. Diagrama de secuencia. Gardar Interfaz

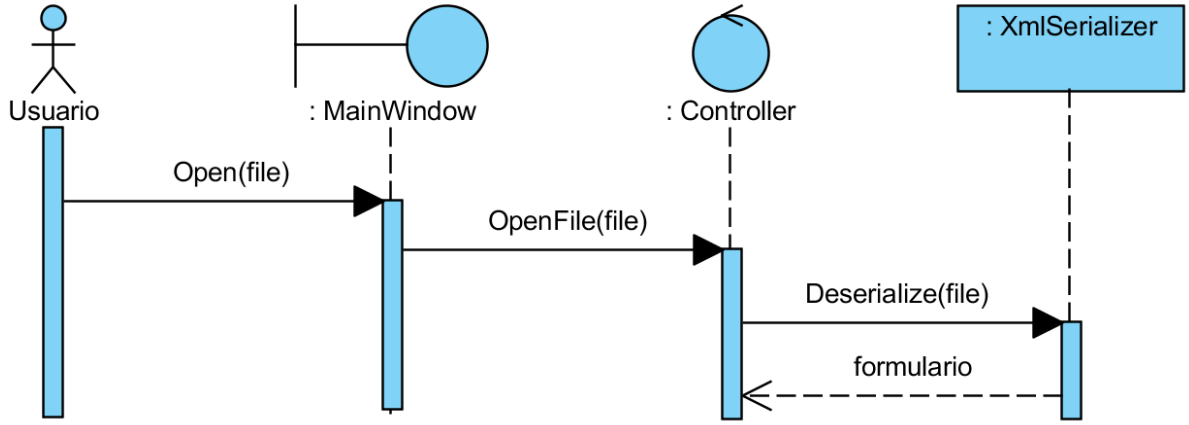

Figura 28. Disagrama de secuencia. Abrir interfaz

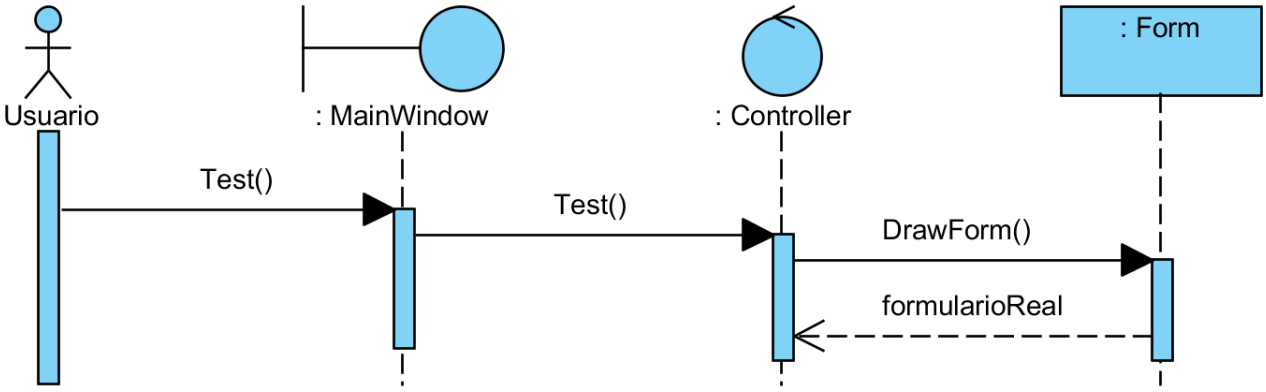

Figura 29. Diagrama de secuencia. Probar interfaz

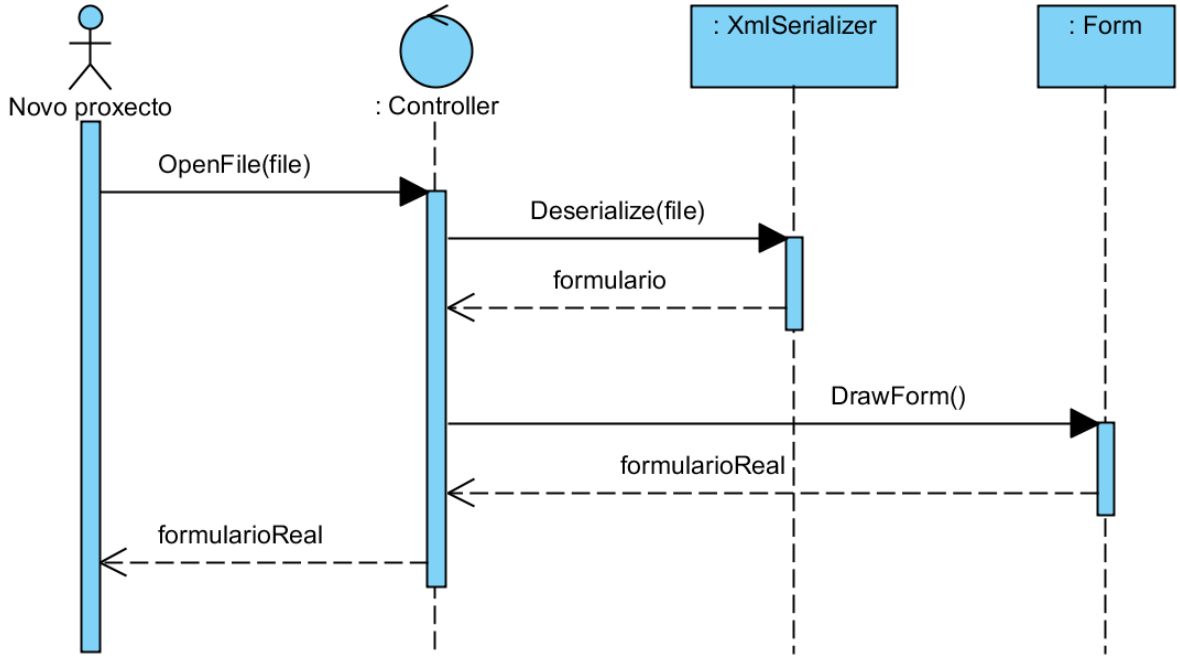

Figura 30. Diagrama de secuencia. Importar interfaz noutro proxecto

Tras obseervar estes diagramas e comprender mellor as necesidades do sistema para dar soporte ás funcionalidades que se van a incluir neste prototipo, desenvolveuse un diagrama de clases tanto para a ferramenta de deseño, na Figura 31, como para a biblioteca para facer as importacións noutro proxecto, Figura 32.

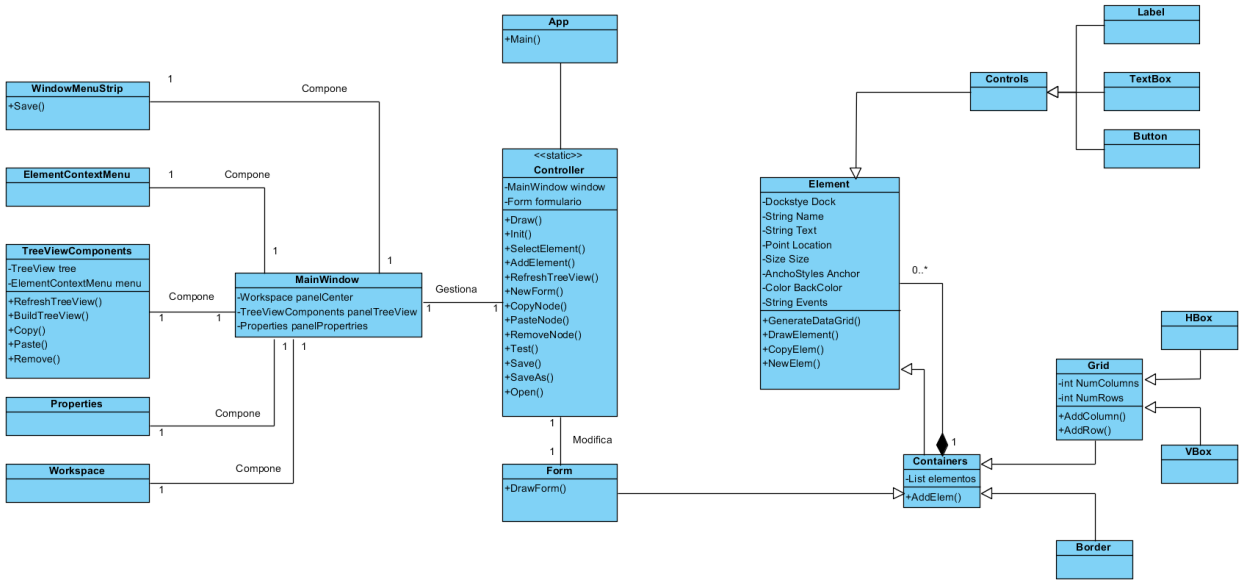

Figura 31. Diagrama de clases do Segundo Prototipo. Ferramenta de deseño

WindowMenuStrip

Esta clase representa ó menú superior da ventana e contén botóns para executar as funcions de crear novo formulario, gardar e abrir unha interfaz, copiar,cortar e pegar elementos e probar e parar proba da interfaz.

Form

Neste prototipo incluese unha función fundamental para ás funcionalidad de proba da interfaz, *DrawForm*. Con esta función crease un formulario da API Winforms con todos os controles equivalentes ós elementos que se lle engadiron coa ferramenta. Os eventos son *Strings* que conteñen a acción a realizar cando se dispare o evento.

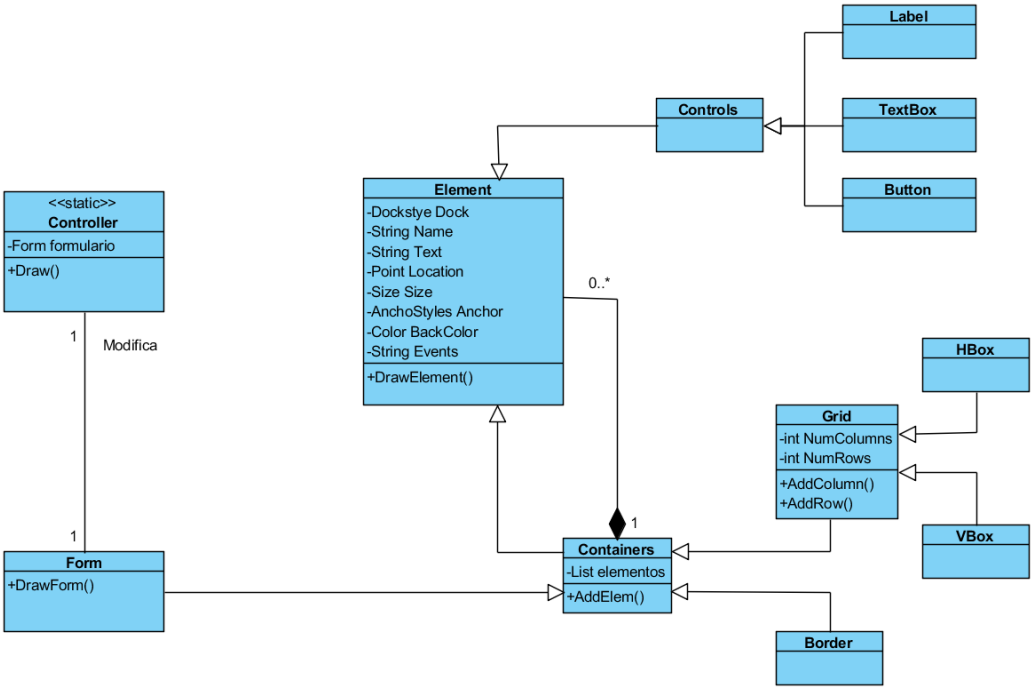

Figura 32. Diagrama de clases do Segundo Prototipo. Bibliteca de importación

Como se pode apreciar no diagrama de clases da biblioteca para importar interfaces nun proxecto, a súa estructura é idéntica a da ferramenta de deseño. A gran diferencia é a carencia de interfaz gráfica e as clases únicamente teñen as funcions de debuxado, é dicir, as funcións que xerán os controles da API Winforms.

4.2. Construcción do prototipo

Nesta ocasión,en canto á ferramenta de deseño, a construcción do prototipo parte do prototipo anterior. Nel hay que engadirlle as novas clases e funcionalidades que se indicaron na fase de deseño. Nesta ocasión engádeselle un menu superior con botones para executar certas accións no sistema, tales coma copiar, pegar, gardar, etc. Ademais tamen se modifica o panel do merxen da dereita para engadir un novo DataGridView que conten os eventos que se permitirán crear coa ferramenta. Tamén engádese a posibilidade de gardar e de abrir interfaces creadas coa ferramenta a través de arquivos .xml. Estes arquvos xeráronse gracias a clase XmlSerializer de C#. Esta permite gardar ou ler os valores das propiedades e atributos de cada un dos elementos que forman a interfaz. Por utlimo neste prototipo impleméntase a funcionalidade que permitirá probar a interfaz que se deseñou. Isto faise creando un formulario de Winforms cos controles equivalentes os elementos do noso sistema, e visualizalo creando unha nova ventana aparte da nosa ferramenta.

Con respecto a biblioteca para importar as interfaces nun proxecto externo créase unha nova solución independente da ferramenta de deseño, e créase unha .dll para que poida ser incluida como referencia noutro prototipo e poder importar as interfaces deseñadas. Ó código desta biblioteca e case totalmente igual ó código da ferramenta de deseño eliminando todas as calses que representan á interfaz gráfica e eliminando todos aqueles métodos ou funcións que non son executados para a creación dun formulario de Winforms.

Como parte final desde prototipo desenvolveuse unhas pequenas probas de funcionamento do sistema de guarda e carga de interfaces deseñadas comprobando que tras engadir unha serie de elmentos ó fomrulario ou a outro contenedor, cando se proba a interfaz esta conteña os elementos que se quera que tiveran.

4.3. Evailuación e retroalimentación do prototipo

Neste segundo prototipo xa se pode apreciar perfectamente como será o sisteman final. Na reunión co tutor para a segunda demostración do sistema observaronse poucos detalles para mellorar. O mais destacable en canto a melloras é que certos *Containers* deberían inicializarse cun valor para a propiedade *Dock* concreta, o *Border* debe iniciarse con esa propiedade a *Fill* o mesmo que o *Grid* e

o *VBox,* e o *HBox* debe iniciarse co valor *Top.*
Decidiuse que no seguinte prototipo deberia permitirse a creación de eventos nos elementos que se engadan na interfaz, polo que a biblioteca tamén debe soportar esta nova funcionalidade.
Ademais é o momento de ampliar o catálogo de elementos a engadir de forma que xa se incluan todos os elementos que se especificaron nos obxetivos do proxecto.
En cuestión de novas funcionalidades neste prototipo xa se cubriron todas as que se planificaran neste proxecto polo que o seguinte prototipo simplemente é para ampliar a variedade de controles e engadir algunha propiedade os elementos xa existentes.

5. Tercer prototipo

5.1. Deseño rápido

Este prototipo ten como obxetivo principal completar os obxetivos do proxecto que faltaban, estes son, completar o catálogo de elementos e permitir o control de eventos dos controles. Cubrir estes novos engadidos fai que a estructura da aplicación creza considerablemente xa que canda control novo supón unha nova clase. Ademais hai que ter en conta que existen unha serie de elementos un tanto especiais, algúns deles, a pesar de non ser elementos contenedores, poseen subelementos. Estas ampliacións da estructura do sistema pódense apreciar na Figura 33.

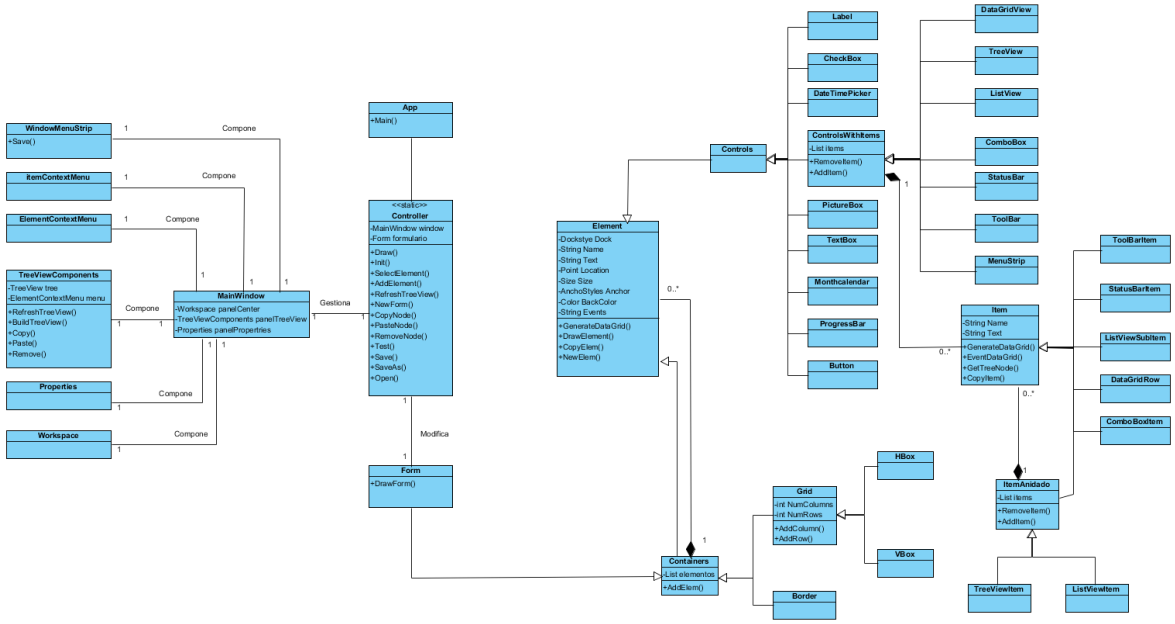

Figura 33. Diagrama de clases do Terceiro prototipo. Ferramenta de deseño

Element

Nesta clase incluese as propiedades correspondentes ós eventos. No diagrama se indican como *String events*, ainda que en realidade son unha serie de *Strings* sendo un por cada evento soportado.

- **Atributos**
  - Conxunto de *Strings* que almacenan as instruccións que se executarán ó lanzarse o evento correspondente
- **Métodos**
  - *EventDataGrid*: Esta función xerá a táboa na que se poderá indicar as instrucións a realizar no evento indicado.

ControlsWithItems

Esta clase reperesenta os elementos que posuen subelementos pero sen ser elementos contenedores. Por exemplo é o caso do *TreeView* que posee nodos. Desta clase extenden os elementos **DataGridView**, **TreeView**, **ListView**, **ComboBox**, **StatusBar**, **ToolBar** e **MenuStrip**.

- **Atributos**
  - *List<Item> items*: Lista que contén os items que forman parte do elemento.
- **Métodos**:

    - *AddItem*: Función que engade un subelemento ó elemento en cuestión.
    - *RemoveItem*: Función que elimina un subelemento do elemento seleccionado.

Item

Esta clase fai referencia ós subelementos que forman parte de controles que extenden da clase *ControlsWithItems*. Desta clase extenden as clases **TooBarItem**, **StatusBarItem**, **ListViewSubItem**, **DataGridRow** e **ComboBoxItem**.

- **Atributos**
    - Os atributos desta clase correspóndense coas propiedades dos Controles da API Winforms equivalentes ós *Items*. Neste caso tan só se permite editar o *Name* e o *Text*.
- **Métodos**
    - *GenerateDataGrid*: Con esta función xérase a táboa onde se poden editar as propiedades dos *Items*.
    - *EventDatagrid*: Esta función xera a táboa onde se poden editar os eventos dos *Items*.
    - *GetTreeNode*: Esta función servirá para xerar a sección de TreeView correspondente ó *Item* no panel esquerdo da interfaz da ferramenta.
    - *CopyItem*: Con este método se poderá copiar un *Item* que forma parte dun *Element* de forma que cando este se copie se copie con todos os *Items* que o forman.

ItemAnidado

Esta clase representa a aqueles *Items* que á sua vez conteñen items tamén, como poden ser os nodos do *TreeView* que tamen poden ter subnodos. Desta clase extenden os *Items* **TreeViewItem** e **ListViewItem**.

- **Atributos**
    - *List<Item> items*: Lista que contén os I*tems* que forman parte do elemento.
- **Métodos**:
    - *AddItem*: Función que engade un subelemento ó elemento en cuestión.

En canto á estructura da biblioteca para a importación de interfaces, xeradas coa ferramenta de deseño, noutro proxecto en desenvolvemento, pódese ver no diagrama de clases da Figura 34 como é idéntica á estructura da ferramenta de deseño, tendo coma diferencia a inexistencia de interfaz gráfica e de métodos que non sexan para xerar os controles da API Winforms equivalentes ós elementos.

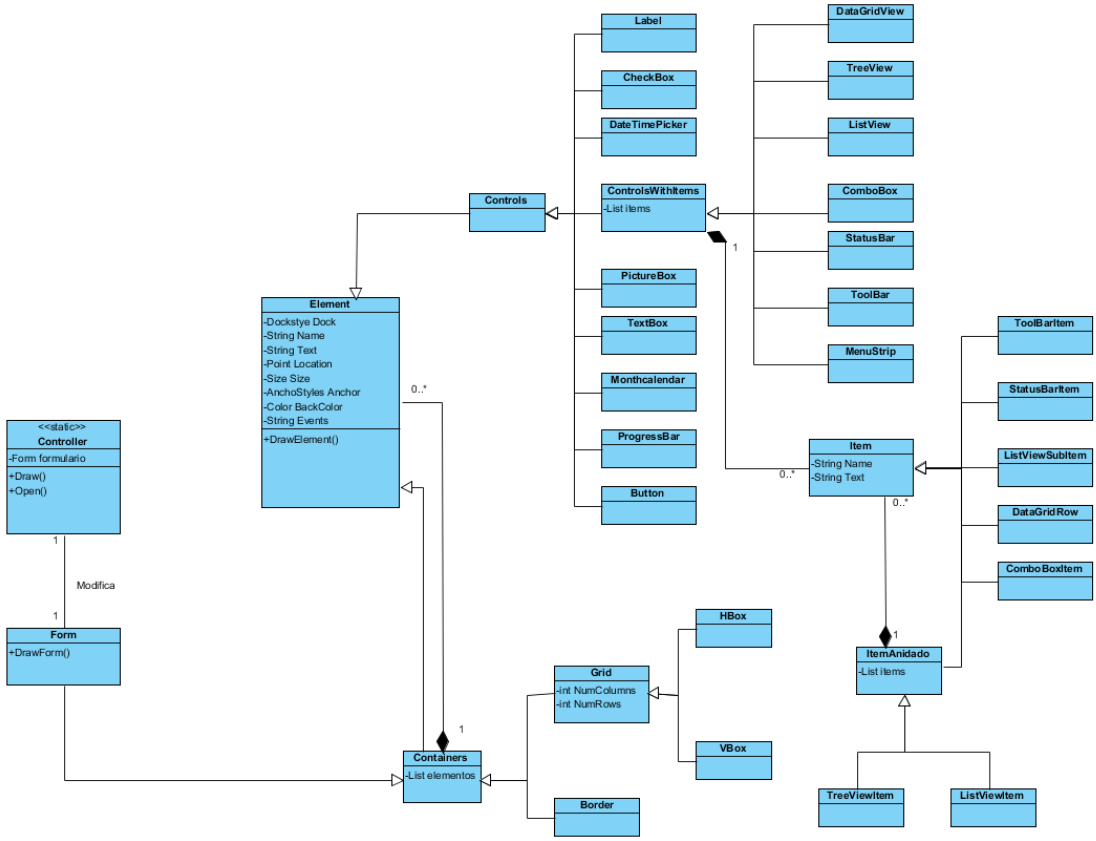

Figura 34. Diagrama de clases do Terceiro prototipo. Biblioteca par importar interfaces

5.2. Construcción do prototipo

Finalmente imos construir o último prototipo planificado neste proxecto. Nesta ocasión aumentou o número de elementos que se lle poden engadir a unha interfaz deseñada con esta ferramenta, polo que a implementación deste prototipo consistitu maioirmente en crear novas clases e en implementar os seus respectivos tests para comprobar que os controles xerados coa ferramenta son tal e como se deseñaron.

5.3. Evaluación e retroalimentación do prototipo

Finalemte faise a reunión de evaluación do terceiro prototipo. Nela compróbase o funcionamento da ferramenta de deseño para estar seguro de que cubre todos os obxetivos e funcionalidades que se esperan do proxecto. Efectivamente así é, o mesmo que a bibliteca para a importación de interfaces en un proxecto C# axeo a este sistema.

Todas as demostracións de funcionamento do sistema fixéronse baixo un sistema operativo GNU/Linux e en ningunha ocasion ata agora foi nun sistema Windows. Tras probalo neste sistema démonos de conta que non funcionaba da forma que se esperaba polo que era necesario correxir certos detalles da implementación que facian que o sistema desenvolvido fora incompatible conWindows. A pesar deste problema non se cree necesario a creación dun novo prototipo senon que tan só é preciso unha pequena sesion de refinamento de cara a ultima fase do proxecto, o desenvolvemento do producto.

6. Desenvolvemento do producto

Tras correxir os pequenos problemas encontrados na demostración do terceiro prototipo, para facelo compatible ó 100% con Windows e GNU/Linux simultaneamente, e como xa se cubriron todos os obxetivos e funcionalidades planificadas para o proxecto, é o momento de convertir o último prototipo no producto final.