# [ PROTOTIPO: 3 Fases ]

### [ LEYENDA ]

**Fase inicial**: Objetivos que deberíamos cumplir en unas 2-5 semanas.

**Fase avanzada**: Objetivos a cumplir dentro del período de entrega.

**Fase extendida**: Objetivos que están **fuera del alcance del proyecto** debido al corto plazo de tiempo.

## Visión general

Nuestro objetivo principal tendría que ser el desarrollo de un conjunto de componentes que puedan ser añadidos a cualquier proyecto desarrollado en JUCE, particularmente, Introjucer. No podemos desarrollar estos componentes dentro del proyecto Introjucer ya que éste actualiza su código fuente con frecuencia y tendríamos que estar siempre adaptándonos a las nuevas versiones. De modo, que partiremos de un proyecto vacío con una ventana y su correspondiente contenedor de componentes. En este contenedor de componentes tenemos que inicializar nuestro editor y llamar a sus mètodos para obtener los distintos componentes necesarios para el editor, como las toolbox o la ventana de propiedades de un componente.

Entonces, tenemos un contenedor de componentes en nuestro programa principal, que es quien inicializa el editor, no el propio editor, su constructora tiene que ser parecido a:

//Inicializamos el editor

Editor editor = new Editor();

//Obtenemos una toolbox

Toolbox \*toolbox = editor->createToolbox(...);

//Añadimos componentes a la toolbox

toolbox->addItem(“juced\_Window”, “Window”, BinaryData::window\_png, ...);

toolbox->addItem("juced\_Label", "Text label", BinaryData::label\_png, ...);

...

//Añadimos la toolbox al contenedor de componentes actual

this->addAndMakeVisible(toolbox);

Y lo mismo vendría a ser para la ventana de propiedades, tiene que ser obtenida a través del editor para luego facilitar al desarrollador que incorpore el editor en su aplicación, así podrá poner dichos componentes en donde él crea que tienen que ir y no necesariamente dentro del componente editor.

## Características [Editor]

1. **Toolbox**: Permite seleccionar una herramienta que corresponde a un componente para ser dibujado en el editor.
2. **Interacción con el editor**:
   1. Pinta el área que va a ocupar el nuevo componente mientras el usuario lo dibuja en el editor. Crea un nuevo componente dentro del componente en el que empezó el drag&drop.
   2. Selecciona un componente en el editor para poder editar sus propiedades, redimensionar y desplazar..
3. **Propiedades**: Permite modificar los atributos de un componente
4. Componentes capaces de ser dibujados por el editor:
   1. **Window**
   2. **Label**
   3. **Textbox**
   4. **Button**
   5. **Listbox**
   6. **Combobox**
   7. **Tabs**
   8. **ImageButton**
   9. **TreeView**
   10. **Slider**
   11. **Viewports**
   12. **Tables**
5. **Historial de cambios**: Permite undo/redo mediante el componente ValueTree.
6. **Exportar**: Guarda la estructura de componentes y sus atributos en un archivo XML.
7. **Importar**: Carga la estructura de componentes y sus atributos de un archivo XML.
8. **Generar código**: genera los archivos de código fuente necesarios para poder compilar la interfaz.
9. Formas de editar el código por parte del usuario:
   1. **Gestionar eventos**: Proporciona funciones en archivos de código fuente separados del código generado para el tratamiento de eventos.
   2. **Edición directa de código**: Permite al usuario añadir líneas de código dentro del código generado, manteniendo un constante registro de cambios.
10. **Componentes externos**: Permite al usuario añadir librerías externas (\*.dll) al componente editor.

## 

## Características [Aplicación]

Hay otras características que no forman parte del componente editor sino de la aplicación que alberga el editor, Introjucer:

1. **Plantillas**: Permite seleccionar un modelo de aplicación específico al crear el proyecto.
2. **Creación de componentes**: Permite crear un proyecto componente para ser compilado como una librería externa (\*.dll).
3. **Añadir componentes personalizados**: Permite al usuario incorporar librerías externas de componentes en el editor, así como mostrar su herramienta para dibujarlos en una de las toolbars.
4. **Interfaz para interactuar con el editor**: Proporciona componentes que permiten importar, exportar, gestionar los atributos del editor (p.e. ancho de la rejilla de fondo).