**Manual de Usuario: DYNAMIC FORMS**

**Introducción**

Dynamic Forms es un framework desarrollado en Python-Django pensado para la generación de formularios dinámicos multi-página y multi-caminos. La creación de dichos formularios se realiza mediante una aplicación web con una interfaz amigable la cual se ha probado que funciona correctamente para las últimas versiones de los navegadores Chrome, Mozilla Firefox e Internet Explorer.

El desarrollo de la aplicación se realizó de manera que pueda ser extensible desde varios puntos de vista, logrando de esta manera obtener nuevos tipos de campo de manera rápida.

**Componentes de un Campo**

Todos los tipos de campo pertenecientes al framework están conformados por una serie de componentes específicos. Dichos componentes son:

* Clase Python:

Todo tipo de campo soportado por la aplicación debe tener una clase Python asociada. Dicha clase debe extender la clase abtracta Field (ubicada en el archivo Field.py dentro del directorio fieldtypes). En dicha clase se definen los diferentes métodos y operaciones soportados por el campo. Entre ellos se encuentran validaciones, operaciones de estadísticas, etc.

También se definen atributos que contienen las rutas donde se ubican los templates html, javascript y CSS asociados al mismo.

* Archivos javascript:

Cada campo puede poseer 3 archivos javascript asociados al mismo.

1. El constructor del Json.

Ubicado en la carpeta static/js/fields. Es un js obligatorio para todo tipo de campo. Contiene una clase que extiende la clase FieldBase, la cual contiene los atributos campo.

1. Operador:

Ubicado en la carpeta satatic/js/operators. Este javascript solo debe incluirse para aquellos campos que tengan operadores lógicos definidos.

Contiene una clase que extiende la clase OperatorField. Todos los métodos que se incluyan a dicha clase se listaran como operadores disponibles.

1. Validator:

Se ubica en static/js/validators. Este javascript solo debe definirse para aquellos campos a los que se les desee agregar algún tipo de validación en el front-end.

* Templates:

Cada campo debe tener 3 templates. Los mismos se ubican en la ruta templates/fields/<nombre\_campo>/

1. properties.html

Genera en la paleta de edición la opciones de validación asociadas al campo.

Se extiende a partir de field\_properties\_base.html

1. template.html

Contiene el código html asociados al campo según será mostrado para el usuario final que complete el formulario.

Se extiende de field\_template\_base.html.

1. template\_edit.html
2. Contiene el código html asociados al campo según será mostrado en la pantalla de edición del framwork.

Se extiende de field\_edit\_template\_base.html.

* CSS

Se ubican en la carpeta static/css/fields.

Contienen los estilos definidos para el campo al que están asociados. Se aplicaran al template.html anteriormente detallado.

**Fábrica**

El framework utiliza los campos definidos de manera dinámica. Esto se logra mediante la utilización de fábricas tanto en el back-end como en el front-end.

Para que un campo pueda ser soportado en la aplicación debe estar registrado en las respectivas fábricas, sino el mismo no será reconocido por la misma. Existe la restricción de que el campo debe registrarse con el mismo nombre en todas las fábricas.

**Crear un nuevo tipo de campo**

Crear un nuevo tipo de campo en el framework Dynamic forms es una tarea muy sencilla y solo requiere realizar los siguientes pasos:

Crear los componentes obligatorios detallados en la sección de **Componentes de un Campo**.

Agregarlos al settings del proyecto.

FIELD\_FILES=(

#otros fields,

<ruta\_a\_la\_clase\_python>

)

**El campo Model Field**

La aplicación Dynamic Forms cuenta con un tipo de campo dinámico que permite generar un combobox cuyas opciones son los datos pertenecientes a un modelo de la aplicación.

Para poder explotar esta característica simplemente es necesario definir un nuevo tipo de campo que derive de la clase abstracta **ModelField**. Luego simplemente es necesario re-definir el atributo model con el modelo deseado y ya se dispondrá del mismo para utilizar en la aplicación.