**Proyecto TREA**

**Registro de Versiones**

**Versión 4.1**

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 28/09/2014 | 1.0 | Especificaciones de producto del primer Sprint | Federico Martínez |
| 28/09/2014  | 1.1 | Verificación SQA | Ari Chamlian |
| 11/10/2014 | 2.0 | Especificación de producto del tercer Sprint | Federico Martínez |
| 11/10/2014 | 2.1 | Verificación SQA | Ari Chamlian |
| 26/10/2014 | 3.0 | Especificación producto del cuarto sprint y resumen administración de código | Federico Martínez  |
| 26/10/2014 | 3.1 | Verificación SQA | Ari Chamlian |
| 09/11/2014 | 4.0 | Especificación producto del quinto sprint y resumen administración de código | Federico Martínez  |
| 09/11/2014 | 4.1 | Verificación SQA | Ari Chamlian |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contenido**

[**1.** **COMPONENTES**](http:///C%3A/LUCIA/ma07/plant/configuracion/SCMRVGXvY.doc) [**3**](#h.1fob9te)

1.1. DynamicForms [3](#h.3znysh7)

1. **Componentes**

* 1. **DynamicForms**

Aplicación web en base a la tecnología Python/Django para la generación de formularios dinámicos para INEFOP.

Provee mecanismos para la generación, modificación, publicación y visor de formularios.

**Línea de trabajo del producto:** Requerimientos

**Versión:** 1.0

**Fecha de la versión:** 14/09/2014

La primera versión del producto se corresponde con el primer Sprint del proyecto. En el mismo se implementaran un subconjunto de funcionalidades del conjunto total de las mismas.

A continuación se enumeran las principales funcionalidades que proveerá la primera versión del producto.

El administrador de INEFOP tiene la capacidad de loguearse dentro de la aplicación, lo que le permitirá ejecutar una serie de historias para cumplir con sus requerimientos. Dentro de estas historias se encuentran:

* Crear nuevo formulario: El administrador puede crear un formulario, a partir de uno vacio. Se cuenta con la posibilidad de agregar nuevos campos, modificarlos o eliminarlos. En esta versión, solo se pueden agregar campos numéricos, de texto, y text areas. El usuario además tiene la posibilidad de especificar si el campo agregado es requerido o no.
* Publicar un formulario: Luego de crear un formulario, el administrador puede publicar un formulario con la posibilidad de que un usuario de INEFOP pueda completarlo con datos en el futuro.
* Modificar un formulario: El administrador, mientras está creando el formulario, tiene la posibilidad de modificar los campos, cambiando sus atributos principales, como el texto, si el campo es requerido o no, y el tipo de campo.
* Completar Formulario: Un usuario tiene la posibilidad de ver un formulario ya creado, a partir de su URL, y completarlo con los datos que proporcionara.

Referencias: Sprint 1-Historias.

**Línea de trabajo del producto:** Requerimientos

**Versión:** 2.0

**Fecha de la versión:** 28/09/2014

La segunda versión del producto se corresponde con el segundo Sprint del proyecto. En el mismo se seguirá con la implementación de las historias que se encontraban en el backlog.

A continuación se enumeran las principales funcionalidades que proveerá la primera versión del producto.

Se amplía la funcionalidad de la aplicación en desarrollo. La siguiente versión a liberar al cliente cuenta con el siguiente conjunto de funcionalidades:

* El administrador tiene la capacidad de agregar o quitar un campo de tipo lista de items a un formulario que está editando, E-mail y cedula.
* El mismo también puede guardar un draft con la con la finalidad de continuar con la edición en otro momento sin lógica y además continuar con la edición del mismo en otro momento.
* Se pueden crear nueva versiones de un formulario publicado, y crear un formulario duplicando la última versión publicada de uno existente.
* El administrador puede reordenar los campos del formulario que está creando y configurar las propiedades de cada campo de acuerdo a validaciones.
* El administrador puede descartar un formulario en edición, cambiar el nombre de un formulario, ver la lista de formularios creados y ver las respuestas de los formularios publicados.

Referencias: Sprint 2-Historias.

**Línea de trabajo del producto:** Requerimientos

**Versión:** 3.0

**Fecha de la versión:** 11/10/2014

La tercera versión del producto se corresponde con el tercer Sprint del proyecto. En el mismo se implementaran un subconjunto de funcionalidades del conjunto total de las mismas.

A continuación se enumeran las principales funcionalidades que proveerá la tercera versión del producto, en adición con las funcionalidades de los sprints anteriores.

* Se podrán eliminar paginas con todo su contenido del formulario que se está editando
* Se incorpora la capacidad de agregar condiciones sobre campos, que determinan si los mismos pueden ocultarse o visualizarse durante el completado de un formulario por parte de un cliente de INEFOP.
* Visualización de estadísticas graficas para su posterior análisis por el administrador de INEFOP.
* Se podrán descartar cambios hechos a un draft.
* Se incorporan filtros para la búsqueda de estadísticas.
* Se podrá ordenar la lista de formularios de acuerdo a criterios preestablecidos (fecha, titulo, id, autor, etc.).
* A los campos se les podrá configurar un tooltip (que muestra información sobre el campo que se está completando)

Referencias: Sprint 3-Historias.

**Registro De Versión Sprint 4**

La tercera versión del producto se corresponde con el cuarto Sprint del proyecto. En él se continuó con la incorporación de las funcionalidades pactadas con el cliente.

**Línea de trabajo del producto:** Requerimientos

**Versión:** 3.0

**Fecha de la versión:** 26/10/2014

Como nuevas funcionalidades se agrego al producto la posibilidad de visualizar datos estadísticas con sus respectivas gráficas. Además se provee al usuario de la funcionalidad de poder exportar a pdf o formato csv las estadísticas para poder procesarlas con otra herramienta.

**Línea de trabajo del producto:** Arquitectura

**Versión:** 3.0

**Fecha de la versión:** 26/10/2014

En la parte arquitectónica de la aplicación, se intento refactorizar el código del front end tanto como el back end para cumplir con el requerimiento del cliente consistente en que el proyecto final será un framework para la creación de formularios web.

Además el cliente espera el desglose de la aplicación y framework. En concreto, se espera contar con una aplicación que consuma los servicios del framework dynamicForms. Actualmente no había una separación clara de los mismos (la aplicación y el framework estaban unificados).

**Administración del código Sprint 4**

Se pusieron a disposición diferentes branches para la realización de este sprint los cuales se enumeran a continuación:

* **Estadística**: en esta rama se implementará la funcionalidad de las estadísticas.
* **Dual-Project**: da soporte al requerimiento del cliente pertinente a la separación de la aplicación y del framework. Más específicamente, en esta rama se encuentra una aplicación que consume los servicios prestados por el framework dynamicForms para la creación de formularios.
* **FrontTesting**: Se comenzaron con los testing en el front-end los cuales continuarán hasta el final de desarrollo.
* **TestingDevel4.0**: rama para utilización del responsable verificación y de sus asistentes que testearan las funcionalidades para este sprint.
* **Devel4.0**: rama estable del desarrollo de la aplicación la cual integrará las nuevas funcionalidades implementadas en las otras ramas.
* **Exportación:** rama para dar soporte a la implementación de exportación a distintos formatos.

**Registro De Versión Sprint 5**

La quinta versión del producto se corresponde con el quinto Sprint del proyecto. En él se continuó con la incorporación de las funcionalidades pactadas con el cliente y además se ahondó en el testeo del código del back end y front end.

**Línea de trabajo del producto:** Requerimientos

**Versión:** 5.0

**Fecha de la versión:** 9/11/2014

Las historias que se implementaron para este sprint son las siguientes:

* Generación en formato pdf de las respuestas de un formulario y también en formato cvs
* El usuario puede agregar “captcha” para verificar que el usuario que esta contestando el formulario es humano
* Se agrego un tipo nuevo de campo File para poder subir archivos.
* Se agrega un tipo nuevo de campo GeoLocation, para dar la posibilidad al usuario de indicar su ubicación en un mapa de google.
* Se agrega un tipo nuevo de campo FieldModel. El mismo es un tipo especial de campo que permite vincular modelos de django/python directamente con la API dynamicForms.
* Se dispone al usuario la posibilidad de suscribirse a signals y de disparar acciones determinadas.

**Línea de trabajo del producto:** Verificación

**Versión:** 5.0

**Fecha de la versión:** 9/11/2014

Se implementó test unitario en front end utilizando el framework de testeo Karma para angular. Además se continuó con el testeo en el back end utilizando el framework de python para test unitario. Se realizaron extensivos test funcionales utilizando la herramienta Selenium. De esta manera se cumplió con las expectativas que el cliente esperaba en cuanto a testing.

**Línea de trabajo del producto:** Implementación

**Versión:** 5.0

**Fecha de la versión:** 9/11/2014

Para este sprint se tuvieron que tener en cuenta varios aspectos referentes a la implementación:

* La posibilidad de utilizar QuerySet que provee django.
* Se realizaron diversos fixes en algunos archivos como urls.py, asociación de nombres a url ( reverse url), aplicacion de collect static
* Se implementa la validación de la lógica en el back front.
* Certificado de respuesta de un formulario.
* En el back end, se deja de acceder directamente a estructuras JSON para mayor prolijidad.
* Se re-factoriza el back end. En especial se redefinieron herencias de templates.