**Documentación**

**Especificaciones técnicas**

**Aplicación móvil**

* Node.js v8.10.0
* IONIC v3

**Aplicación web**

* Python +2.7.x
* Django v1.9.13
* pip v19.1.1
* **Módulos de Django requeridos:**
* django
* djangorestframework
* psycopg2
* django-webpack-loader
* python-dateutil
* XlsxWriter
* django-import-export
* requests
* **Base de datos**
* PostgreSQL v9.4.22

**Credenciales:**

**Aplicación móvil:**

usuario - PruebaINAMHI

**Aplicación web:**

**URL:** http://150.136.230.16/

inamhi - 1nAmh12019

**Administrador de Django:**

insavit - In5av1t123

**CKAN:**

**URL:** http://150.136.213.20

danijo1995 - Food1234

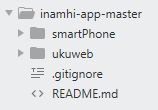
Para acceder al servidor se debe poner los archivos id\_rsa y id\_rsa.pub en un directorio como el siguiente: /home/<usuario>/.ssh/ y desde allí ejecutar en la consola de comandos:

ssh opc@150.136.230.16 -i /home/<usuario>/.ssh/id\_rsa

**clave:** Uku123

**Directorio del código**

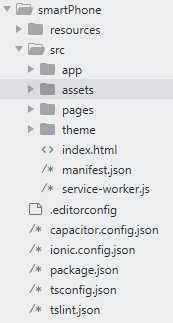
El código está dividido en smartPhone, que contiene el código de la aplicación móvil y ukuweb, que contiene el código de la aplicación web.



**Código de la aplicación móvil**

Dentro de la carpeta smartPhone se encuentra la carpeta src que contiene el código modificable de la aplicación. Las demás carpetas y archivos son generados por IONIC. Dentro de src están las carpetas:

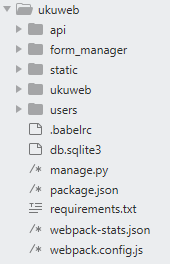
* **app:** Contiene los archivos que se ejecutan al iniciar la aplicación:
* **app.component.ts:** Contiene la lógica que se ejecuta al iniciar la aplicación.
* **app.html:** Contiene la interfaz del menú lateral de la aplicación.
* **app.module.ts:** Contiene los módulos que utiliza IONIC.
* **app.scss:** Contiene las reglas de estilo que se aplican de forma general a toda la aplicación.
* **assets:** Dentro de esta carpeta se encuentran las subcarpetas que contienen las imágenes, íconos y demás archivos estáticos que se utilizan en la aplicación.
* **pages:** Aquí se encuentran las carpetas de las diferentes páginas de la aplicación. Todas ellas contienen 3 archivos: un archivo .ts que contiene la lógica, un archivo .html que contiene la interfaz y un archivo .scss que contiene las reglas de estilo de la página. Las carpetas son:
* **auth:** Página de autenticación del usuario.
* **followUp:** Página para formularios de seguimiento.
* **form:** Página para la presentación de formularios con sus preguntas.
* **formularios:** Agrega los formularios a las listas de formularios pendientes o enviados.
* **home:** Página principal de la aplicación.
* **pendingForms:** Administra los formularios pendientes por realizar y su envío.
* **perfil:** Página del perfil del usuario.
* **sentForms:** Administra los formularios enviados al servidor y su eliminación en la aplicación móvil.



**Código de la aplicación web**

La carpeta ukuweb se subdivide en las siguientes carpetas:

* **api:** Contiene los archivos referentes a validación de usuario, envío de formularios y guardado de formulario.
* **form\_manager:** Contiene los archivos relacionados con el gestor de formularios, sus rutas, modelos, vistas, etc.
* **static:** Contiene los archivos estáticos de hojas de estilo, javascript, imágenes, fuentes de texto, etc.
* **ukuweb:** Contiene los archivos principales que inician la aplicación web. Este módulo invoca a los demás módulos (api, users, form\_manager).
* **users:** Contiene los archivos relacionados con el gestor de usuarios, sus rutas, modelos, vistas, etc.



**Privilegios de los usuarios**

**Administrador**

El usuario administrador, tiene acceso a la aplicación web, desde donde puede realizar las siguientes acciones:

* Crear, ver, editar y eliminar usuarios (administradores y entrevistadores).
* Ver los datos de los formularios que los entrevistadores han realizado.
* Descargar en formato xls, la información de los formularios realizados por los usuarios entrevistados.
* Subir y eliminar plantillas de formularios.
* Ver información de los campos de los formularios.

**Entrevistador**

El usuario entrevistador, tiene acceso a la aplicación móvil, desde donde puede realizar las siguientes acciones:

* Obtener las plantillas creadas por el administrador.
* Llenar, ver, editar y eliminar formularios.
* Ver la lista de formularios en edición y enviados al servidor.
* Permitir la geolocalización del dispositivo.

**Administrador de usuarios**

El usuario administrador puede crear un usuario, editarlo, ver sus datos y eliminarlo. Cuando se crea un entrevistador, automáticamente tiene acceso a todas las plantillas que el administrador ha creado.

**Administrador de plantillas**

El usuario administrador puede crear un formulario, adjuntando una plantilla en formato json con las descripciones descritas en la sección ***Estructura de una plantilla***. Ingresa el nombre de la plantilla, el tipo (SIMPLE o COMPUESTA), selecciona el archivo, la cantidad de formularios a realizar por usuario, el nombre del conjunto de datos de CKAN (será descrito más adelante).

**Formularios**

En la base de datos de la aplicación móvil, se guardan varios tipos de datos:

**- formsData:** Guarda el formulario y sus diferentes versiones con los datos ingresados por el usuario.

**- infoTemplates:** Guarda información de las plantillas, como el identificador (uid), los números de formularios realizados y pendientes y la estructura de la plantilla.

**- pendingForms:** Guarda la información de los formularios pendientes con sus versiones (es una lista de formularios).

**- sentForms:** Guarda información de los formularios enviados (es una lista de formularios).

**Tipos de plantilla:**

* **SIMPLE:** Se utiliza para formularios que se realizan una sola vez y normalmente está dirigido a un solo objetivo. Ej: formularios de satisfacción de un producto comercial.
* **COMPUESTA:** Se utiliza para formularios que requieren un control a lo largo de un tiempo determinado. Ej: formularios relacionados con la salud de una persona. Se dividen en 2 subtipos:
  + **INICIAL:** Se realiza una vez para saber el estado inicial del objetivo.
  + **SEGUIMIENTO:** Se realiza según cierto intervalo de tiempo para analizar la evolución del objetivo.

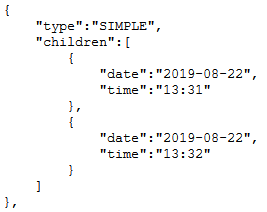
**Estructura del archivo de formulario**

El archivo donde van las preguntas del formulario está dividido en 2 secciones:

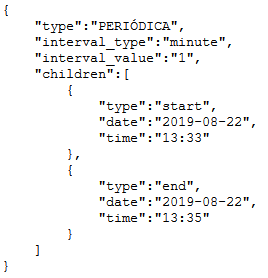
**Notificaciones**

Su etiqueta en el archivo es: “notifications”. Aquí se definen las notificaciones que se presentarán en la aplicación móvil. Existen 3 tipos de notificaciones con diferentes campos y pueden agregarse varios tipos en un mismo formulario:

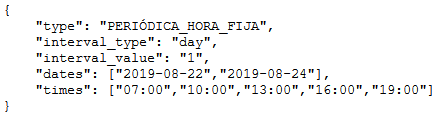
* **SIMPLE:** Permite establecer notificaciones para fechas específicas, agregando solo la fecha y hora de cada notificación.



* **PERIÓDICA:** Permite establecer notificaciones periódicas estableciendo una fecha y hora de inicio y fin y el tipo (minute, day, hour, week, month) y cantidad del intervalo entre notificaciones.

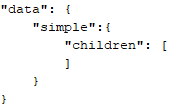


* **PERIÓDICA\_HORA\_FIJA:** Permite establecer intervalos entre notificaciones con horario fijo. Se debe especificar el tipo y cantidad del intervalo. En el campo “dates” se especifican las fechas de inicio y fin en que se mostrarán las notificaciones, mientras que en el campo “times” se ponen las horas específicas en que se mostrarán.

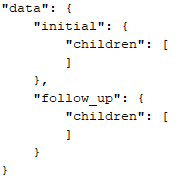


**Secciones y campos**

Su etiqueta en el archivo es: “data”. En esta sección se definen el tipo de formulario, las secciones del formulario y las preguntas que contendrá. El tipo de formulario se escribe dentro de la etiqueta “data” y tiene un campo “children” donde se agregarán las preguntas.



**Ejemplo de formulario simple**



**Ejemplo de formulario compuesta**

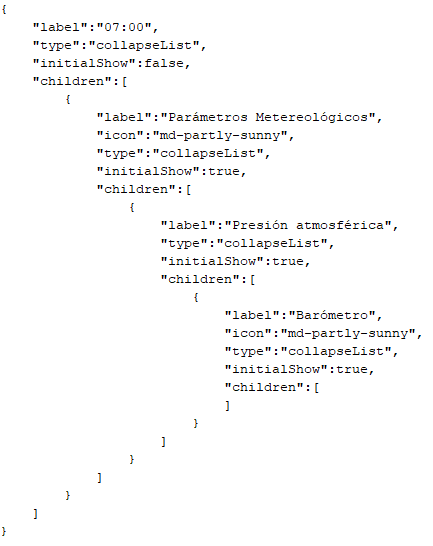
Las secciones y preguntas pueden ser de varios tipos, cada uno tiene ciertas características específicas, pero en general todos tienen atributos comunes:

* **“label”:** Cualquier texto que represente el nombre del campo.
* **“type”:** Este atributo define el tipo del campo con los valores: input\_number (campo numérico), input\_text (entrada de texto), select (lista), checkbox (casillas de verificación), radio (opción múltiple), date o time (fecha u hora).
* **“id”:** Identificador de la pregunta. Debe ser único en el archivo para poder obtener el dato ingresado.
* **“blurFunction”:** Función de validación o cálculo para la pregunta. Se coloca una lista clave-valor con el nombre de la función y los id de las preguntas que requiere la función o se puede dejar vacío.
* **“error”:** Este campo se deja vacío. Aquí se guarda, de existir, el texto de error de la pregunta cuando se haya validado con la función correspondiente del archivo de cálculos.
* **“value”:** Este campo se deja vacío. Aquí se guarda el dato ingresado por el usuario en la pregunta.

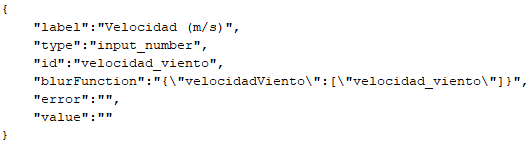
Por otro lado, algunos campos tienen atributos específicos que se deben mencionar:

**Sección:** Una sección puede contener varias subsecciones dentro de ella, pero todas ellas son del tipo “collapseList” y contienen un atributo “children” donde se escriben las subsecciones y preguntas. Además, poseen un atributo “label” o título y un atributo “initialShow” que puede tener como valor “true” o “false”. Este atributo define si la lista de subsecciones se muestra desplegada o recogida.

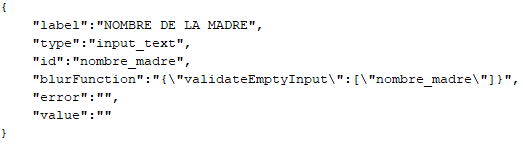
Finalmente, las subsecciones con jerarquías pares poseen el atributo “icon” donde se define un ícono que se mostrará junto a cada elemento de la lista de subsecciones. Estos íconos se pueden encontrar en la documentación de IONIC.



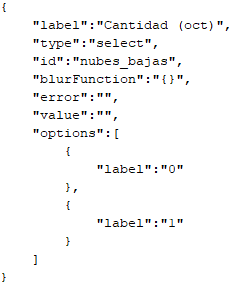
**Campo numérico:** Tiene el tipo “input\_number”. Este tipo de campo permite solo valores numéricos, tanto enteros, decimales y números negativos dependiendo de las validaciones que se le hagan.



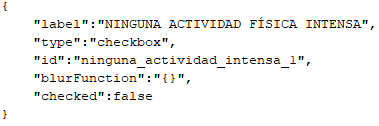
**Entrada de texto:** Tiene el tipo “input\_text”. Este tipo de campo permite ingresar cualquier texto y tiene menos atributos que los demás tipos de campo.



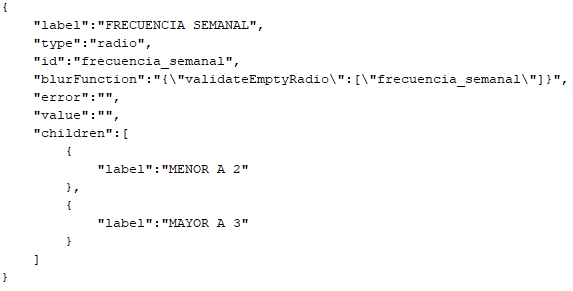
**Lista:** Tiene el tipo “select”. Permite presentar una lista de opciones para que el usuario escoja una sola. Tiene un atributo adicional “options”, donde se colocan las opciones que se mostrarán en la lista.



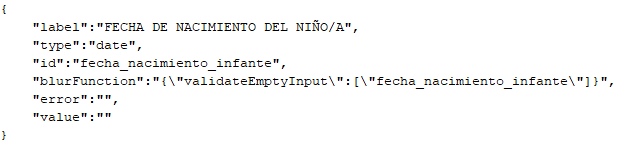
**Casillas de verificación:** Tiene el tipo “checkbox”. Posee menos atributos que los demás tipos de campos, pero tiene un atributo “checked” cuyo valor puede ser “true” o “false”, y define si la opción aparece marcada al mostrarse la pregunta.



**Opción múltiple:** Tiene el tipo “radio”. Permite escoger una sola opción de los elementos listados debajo de la pregunta. Posee el atributo “children” donde se escriben las opciones disponibles para escoger.

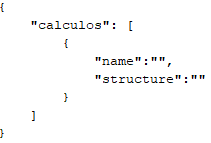


**Fecha/hora:** Tiene el tipo “date” o “time”. Permite escoger una fecha u hora específica.



**Estructura del archivo de cálculos**

El archivo de cálculos tiene como nombre de sección “calculos” y dentro de ella se listan todas las funciones escritas en Javascript que van a ser utilizadas en el formulario. Además, las funciones deben estar escritas en una sola línea, ya que esta línea se toma como un conjunto de caracteres para luego ser ejecutada cuando se ingrese el dato en la pregunta correspondiente.



**CKAN**

**¿Cuándo se envían las plantillas a CKAN?**

Al momento de crear una plantilla en el servidor, se ingresa el nombre del conjunto de datos (dataset) a donde se almacenarán las plantillas en CKAN. Una vez que se crea la plantilla en el servidor, se crea un recurso en CKAN.

**¿Cuándo se envían los formularios llenos a CKAN?**

Cuando un cliente (dispositivo móvil o tablet) envía un formulario, el servidor recibe la información referente a dicho archivo como el nombre, las coordenadas, los datos del formulario, fecha de creación, fecha de envío, código del entrevistado, si incluye gps o no, y lo almacena en la base de datos, después convierte los datos de ese formulario que está en formato json a csv y guarda el archivo en la carpeta www/static/forms y por último obtiene el id del set de CKAN a donde debe ser enviado el archivo y lo envía.

**¿Qué pasa si elimino la plantilla de un formulario desde INSAVIT?**

Si se elimina una plantilla de un formulario desde el servidor, no se elimina del conjunto de datos en CKAN. Los formularios enviados que pertenecen a dicha plantilla tampoco se eliminan. Cuando el usuario entrevistador se autentica, recibe las plantillas que están almacenadas en el servidor, no en CKAN, por lo que, si elimina una plantilla, el usuario ya no tendrá en su dispositivo dicha plantilla, aunque en CKAN sí exista. Si el dispositivo tiene formularios con la plantilla que ha sido eliminada en el servidor, al enviar los datos, éstos se guardan normalmente. Si en la plantilla se incluía el id del conjunto de datos, se envían los archivos en formato csv a CKAN.

**Mejoras para futuras versiones**

**Aplicación móvil**

* Mejorar la navegación entre secciones del formulario. Poner botones para ir a la anterior y siguiente sección.
* Hacer que se descarguen los archivos de plantilla y cálculos del servidor, sólo cuando hayan sido modificados para evitar hacer peticiones innecesarias.
* Ampliar el soporte para otros tipos de notificaciones y formularios.

**Aplicación web**

* Crear grupos de usuario para que únicamente tengan acceso a ciertas plantillas los usuarios seleccionados.
* Agregar un gestor de formularios que permita crear de forma dinámica un formulario, sin tener que crear un archivo JSON.
* Hacer que se puedan editar más atributos de los usuarios y plantillas.
* Hacer que se pueda subir el archivo de cálculos desde la aplicación web.