


# URANUS - Manual de Desarrollador

Bogotá 5 de noviembre de 2018


URANUS - Versión 2.0

Versión Doc. 1.0

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0


## Resumen Ejecutivo

El presente documento trata de una consolidación de los documentos de especificación de requerimientos de software, especificación de arquitectura de software y manual de usuario desarrollador. En la primera sección se describe la visión del sistema en donde se especifican los requerimientos funcionales y no funcionales. En la segunda sección se describe la lógica e implementación del sistema donde se recomienda una estrategia para extensión del sistema. Y por último se encuentra la sección de instalación en donde se especifican los comandos para el acondicionamiento del entorno de desarrollo.

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0

## Historial de Modificaciones


<b>Fecha</b>	<b>Descripción</b>	<b>Autor</b>
05/11/18	Inicio del documento.	Fabián José Roldán

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0

## Visión del Sistema

El objeto de la aplicación es soportar de forma interactiva un proceso de experimentación al recopilar datos de uno varios Resolutores (sujetos de estudio) al momento de interactuar con un problema físico-matemático de proporcionalidad con el objetivo de obtener datos de variabilidad intra e intersujeto respecto del tipo de control de variables y covariación utilizadas. Las principales características de sistema son las siguientes:

- ❖ La aplicación es WEB, lo cual permitir un fácil despliegue y uso de las funcionalidades. Consta de:
  - Un cliente semi-pesado que incluye código *Javascript* y el servidor almacenar los datos en BD o en el *filesystem* (usando archivo Excel) para consignar la información de captura y reportes.
  - La aplicación es sensible al dispositivo (*responsive*).
- ❖ La funcionalidad permite el registro de los usuarios Resolutores capturando los siguientes conjuntos de datos:
  - Nivel de escolaridad (Primaria, Secundaria, Universitaria) a su vez la aplicación permite la selección de la institución educativa de forma geo-localizada en términos de la entidad territorial de nivel y nivel superior (municipio, departamento) y tipo (urbano o rural). Si es primaria o secundaria se pregunta el grado, si es nivel universitario se preguntará la facultad y el semestre.
  - Datos personales con el carácter de no individualizar al Resolutor, tales como: fecha de nacimiento y género.
  - La aplicación genera un ID por cada Resolutor que debe estar compuesta por la fecha de nacimiento, institución educativa y grado.
- ❖ La aplicación consta de tres problemas o tareas: Movimiento lineal, Manejo de fluidos y Equilibrio (balanza). La aplicación prevalece el carácter de que las tareas son isométricas morfológicamente hablando. La aplicación una vez que el usuario se registra le presenta de manera aleatoria las tareas, las tres tareas se deben cumplir a cabalidad y por cada una de las tareas se requiere lo siguiente:

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0


- Ingreso de dos valores a través de un componente grafico (p.e relojes), estos dos valores representan las dos variables en cada problema. En el movimiento lineal es velocidad y tiempo para el desplazamiento. En el de fluidos es apertura de la llave y tiempo para el llenado. En el equilibrio es distancia desde el pivote y peso.
- La aplicación permite definir los puntos de meta (lo define otro usuario - Administrador), para que de esta forma, después de cada intento del Resolutor, se pueda validar que tan lejos está de la meta. Por ejemplo en la tarea1 (movimiento) se define como meta el valor de la distancia.  $x=v.t$ , por lo ejemplo 80. Si el Resolutor asigna 10 a velocidad y 7 al tiempo. Alcanza una distancia de 70, por lo tanto está a 10 de la meta. Esto se puede estipular también en porcentaje (llego al 87.5%).

Como se mencionó anteriormente los valores de los diferentes intentos para cada tarea se almacenan en el servidor en una base de datos Postgres y en el sistema de archivos con formato Excel. Donde en la primera hoja se coloca la información del Resolutor y en la segunda hoja la información de cada intento (variable 1, variable 2, porcentaje de acercamiento a la meta, tiempo para cada intento). A continuación se describen de una manera formal las características del sistema.

## Características Funcionales del Producto

En la siguiente tabla se muestra los requerimientos funcionales que el producto debe satisfacer:

ID	Descripción
RF001	La aplicación debe permitir el registro de los usuarios Resolutores capturando los siguientes conjuntos de datos: <ul style="list-style-type: none"> <li>→ Nivel de escolaridad (Primaria, Secundaria, Universitaria) a su vez la aplicación debe permitir si es primaria o secundaria se preguntará el grado, si es nivel universitario se preguntará la facultad y el semestre.</li> <li>→ Datos personales con el carácter de no individualizar al Resolutor, tales como: fecha de nacimiento y género.</li> </ul>
RF002	La aplicación debe permitir la selección de la institución educativa de forma geo-localizada en términos de la entidad territorial de nivel y nivel superior (municipio, departamento) y tipo (urbano o rural).
RF003	La aplicación debe permitir generar de forma automática un ID por cada Resolutor que debe estar compuesta por la fecha de nacimiento e institución educativa al momento de registro.

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0

RF004	La aplicación debe permitir al Resolutor interactuar con una animación de Movimiento lineal.
RF005	La aplicación debe permitir al Resolutor interactuar con una animación de Manejo de fluidos.
RF006	La aplicación debe permitir al Resolutor interactuar con una animación de Equilibrio (balanza).
RF007	La aplicación debe permitir una vez que el usuario se registra debe presentarle de manera aleatoria las tareas.
RF008	La aplicación debe permitir que las tres tareas se deben cumplir a cabalidad.
RF009	La aplicación debe permitir para cada tarea el Ingreso de dos valores a través de un componente grafico (p.e. relojes), estos dos valores representan las dos variables en cada problema. En el movimiento lineal es velocidad y tiempo para el desplazamiento. En el de fluidos es apertura de la llave y tiempo para el llenado. En el equilibrio es distancia desde el pivote y peso.
RF010	La aplicación debe permitir que un usuario Administrador modifique parámetros de las tareas tales como: <ul style="list-style-type: none"> <li>→ Los puntos de meta.</li> <li>→ Rangos de valores.</li> <li>→ Intervalo de confianza.</li> <li>→ Número de intentos que se le permitirán al Resolutor (inicialmente el valor es de 15 intentos).</li> </ul>
RF011	La aplicación debe informar al Resolutor el progreso tanto de la tarea en particular como del conjunto de tareas.
RF012	La aplicación debe permitir guardar los valores de los diferentes intentos para cada tarea ser almacenados en el servidor en un sistema de archivos tipo Excel. Donde en la primera hoja se coloca la información del Resolutor y en la segunda hoja la información de cada intento (variable 1, variable 2, porcentaje de acercamiento a la meta, tiempo para cada intento).


## Características no Funcionales del Producto

En la siguiente tabla se muestra los requerimientos no funcionales que el producto debe satisfacer:

ID	Descripción
RNF001	La aplicación debe prevalecer el carácter de que las tareas son isométricas morfológicamente hablando.
RNF002	La aplicación debe ser sensible o adaptarse a la resolución del dispositivo.

## Restricciones del Producto

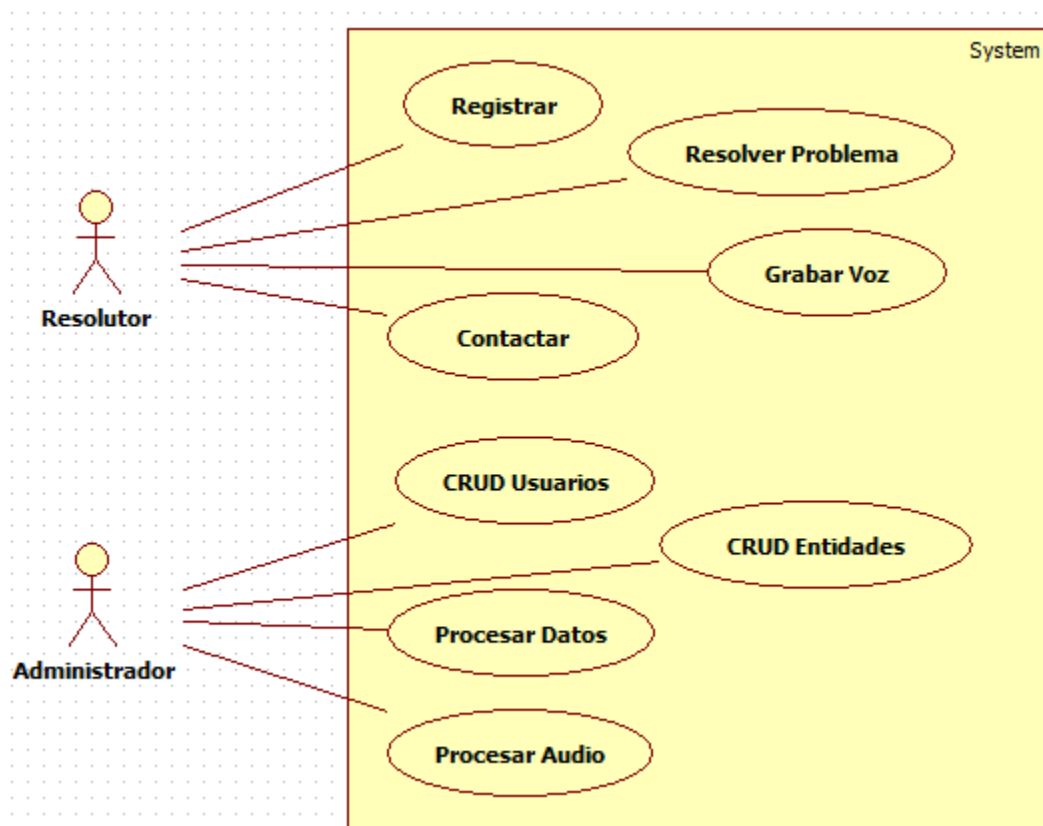
En la siguiente tabla se muestra las restricciones inherentes del producto que debe cumplir:

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0

ID	Descripción
R001	La aplicación debe desplegarse en un entorno WEB definido por la Universidad.
R002	La aplicación debe ser una aplicación de dos niveles.
R003	La aplicación debe permitir un componente cliente semi-pesado para incluir código Javascript.
R004	La aplicación debe permitir que del lado del servidor se almacenen los datos en BD o en el filesystem (usando archivo Excel).


## Casos de Uso

En la Figura 1. Se puede apreciar los principales casos de uso del sistema:



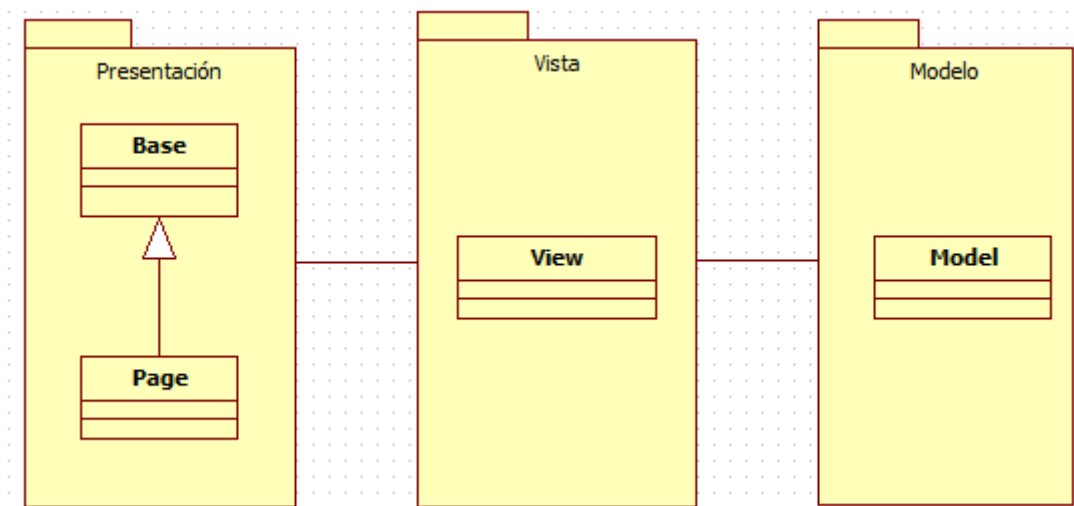
**Figura 1. Casos de Uso del Sistema**

El sistema consta de dos usuarios (Resolutor y Administrador) en donde el Resolutor puede efectuar un registro, resolver problemas, grabar su voz (para las tareas de representación) y enviar un mensaje de contacto a los administradores del sistema. El usuario administrador puede crear, obtener, eliminar o actualizar (CRUD) de usuarios y entidades (Resolutores, Intentos y Problemas). También puede generar la orden al sistema de procesar los datos y el audio.

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0

## Lógica e Implementación del Sistema

Una vez identificadas las características que debe tener el producto en la Figura 2. Se puede apreciar la vista lógica del sistema:




**Figura 2. Vista Lógica del Sistema**

Como se puede apreciar en la Figura 1. El sistema se rige bajo una estructura M (modelos) y V (vistas) donde toda la interacción con los usuarios está regida por los componentes de presentación y vista donde la vista cumple el rol de control. Las vistas se comunican con el modelo que rige una abstracción del modelo de datos de la aplicación (Se usó PostgreSQL versión 10). El sistema fue implementado en el lenguaje de programación Python 3.6.5 sobre la plataforma Anaconda en su versión 1.9.2 utilizando Django en su versión 2.1. Para extender la aplicación se debe comenzar desde las vistas. Como se puede apreciar en la Figura 2. En la presentación existen dos tipos de componentes: un componente base y componentes de página. Se usó el esquema de plantillas de Django<sup>1</sup> por lo tanto el componente Base hace referencia al artefacto 'base.html' y el componente Page hace referencia a las páginas que implementan la plantilla. Django está organizado por módulos, el módulo que se implementó se llama Tasks cada módulo tiene su script para las vistas (views.py) y script para los modelos (models.py) en estos artefactos se encuentran respectivamente las vistas y los modelos tenidos en cuenta para el sistema.

<sup>1</sup> Tutorial de uso de plantillas: <https://djangobook.com/templates-in-views/>



	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0

## Instalación


En la Figura 3. Se puede apreciar las principales librerías que fueron usadas para acondicionar un entorno en Anaconda:

✓ certifi	Python package for providing mozilla's ca bundle.	<a href="#">2018.8.13</a>
✓ django	Web framework that encourages rapid development	<a href="#">2.1</a>
✓ krb5	A network authentication protocol	1.16.1
✓ libpq	The postgres runtime libraries and utilities (not the server itself)	10.5
✓ openssl	Openssl is an open-source implementation of the ssl and tls protocols	<a href="#">1.0.2p</a>
✓ pip	Pypa recommended tool for installing python packages	10.0.1
✓ psycopg2	Postgresql database adapter for python	2.7.5
✓ python	General purpose programming language	<a href="#">3.7.0</a>
✓ pytz	World timezone definitions, modern and historical	2018.5
✓ setuptools	Download, build, install, upgrade, and uninstall python packages	<a href="#">40.0.0</a>
✓ tk	Dynamic programming language with gui elements	<a href="#">8.6.7</a>
✓ vc	A meta-package to track vc features.	14.1
✓ vs2015_runtime	Bundles of the msvc 14 (vs 2015) runtime	15.5.2
✓ wheel	Built-package format for python	<a href="#">0.31.1</a>
✓ wincertstore	Python module to extract ca and crt certs from windows' cert store	0.2
✓ zlib	Unobtrusive compression library	1.2.11

**Figura 3. Componentes de Entorno**

A continuación se especifican los comandos necesarios para crear un entorno Anaconda y desplegar el entorno en un ambiente Linux con todos sus componentes:

```
1720 curl -O https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh
1721 sha256sum Anaconda3-5.0.1-Linux-x86_64.sh
1728 bash Anaconda3-5.0.1-Linux-x86_64.sh
1733 conda create --name uranus_env python=3
```

	<b>AKEN – AKENPAY S.A.S</b>	<b>Proyecto:</b> Cambio Cognitivo.	
	Manual de Desarrollador	<b>Autor:</b> Ing. Fabián José Roldán.	
	Bogotá, Colombia. Lunes 05 de noviembre de 2018.		Versión: 1.0

```

1734 source activate uranus_env
1738 pip install Django
1750 pip install xlswriter
1752 pip install dateutil
1719 pip install psycpg2
1760 pip install python-dateutil
1762 conda update -n base -c defaults conda
1764 conda install -c conda-forge speechrecognition
1766 conda install nltk
1794 cd /etc/
1798 cd sources.list.d
1802 sudo touch pgdg.list
1843 sudo chmod 777 pgdg.list
1844 vi pgdg.list
1845 wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add
-
1846 sudo apt-get update
1847 apt-get install postgresql-10
1848 sudo apt-get install postgresql-10
1905 sudo -u postgres createdb uranusdb
1906 sudo -u postgres psql postgres
1908 python manage.py migrate
1912 pip install psycpg2-binary
1916 python manage.py createsuperuser
1917 python manage.py runserver
1922 python install.py

```