



Guía para el Concursante.

Propósito

Brindar una guía que contenga pautas para el desarrollo del sistema encargado de la automatización y digitalización del proceso de realización de pedidos. Incluye definición de tecnologías y frameworks bases, así como versiones en los casos que se señale, a emplear en la implementación del proyecto así como versiones.

Adicionalmente se definirá la estructura base que tendrá el proyecto y especificaciones en el uso del controlador de versiones a tener en cuenta para el formato que deberá cumplir el entregable que se realizará en cada una de los cortes evaluativos del concurso.

Contenido

[Tecnologías Bases a emplear.](#)

[Estructura del Entregable.](#)

[Código Fuente \(GitHub\)](#)

[Base de Datos](#)

[App Android](#)

[Documentación](#)

[Criterios Para Evaluación y Selección](#)

[Innovación](#)

[Corrección](#)

[Experiencia de Usuario](#)

[Profesionalidad](#)

[Enlaces Útiles](#)

Tecnologías Bases a emplear.

Por necesidades del organizador del concurso se definen las siguientes tecnologías y frameworks para el desarrollo del sistema y su empleo será de carácter obligatorio. A continuación se listan las tecnologías:

- **Django (versión 1.8 +):** Framework basado en Python para el desarrollo de aplicaciones Web que sigue el principio MVC o MTV. En general incluye facilidades para la creación de templates html y jerarquía de estos gracias a su motor de plantilla así mecanismos para la gestión de modelos de datos persistentes y vistas o clases controladores para responder a solicitudes http y https. Se recomienda el uso de la versión 1.8 o superior teniendo en cuenta que esta versión es la de mayor tiempo de soporte en la actualidad.
- **Python (versión 2.7 +):** Versión 2.7 o superiores de Python que tendrá soporte seguro por Django al menos en los próximos 2 años.
- **PostgreSQL (versión 9.X):** Como gestor de BD para el almacenamiento de los datos.
- **Bootstrap 3.x:** Como framework CSS Web Responsive, para el diseño de la interfaz y maquetación del sistema.
- **Controlador de versiones GIT (GitHub):** El código del proyecto así como el entregable deberá estar versionado con el controlador de versiones y contar con las especificaciones que se definen en los criterios para evaluación en el acápite de control de versiones.
En este sentido el equipo de evaluadores del concurso crearán los repositorios para cada equipo. A cada uno se les dará el respectivo acceso y así podrán publicar los proyectos con el código fuente y todos los archivos necesarios.
- **Sistema Operativo (Opcional):** Se recomienda el uso de Linux teniendo en cuenta la compatibilidad de algunas aplicaciones de terceros para Django por lo cual se recomienda Ubuntu 14.04.
- **Android:** Para el caso de las aplicaciones móviles se define el uso de Android como plataforma principal.

Uso exclusivo de librerías, plantillas de diseño y aplicaciones de la comunidad (Open Source o Free):

En este sentido, el uso de librerías de costo por distribución o uso será penalizado, quedando así solo permitido aquellas de código libre o sin costo (open source o free).

NOTA: Cualquier solución que, por algún motivo, necesite configuraciones, instalaciones o softwares adicionales, diferentes o específicas, debe ser comunicado al equipo de Intellisys Challenge para ser evaluado y determinar si es aceptable. Esta comunicación debe hacerse a través del foro del concurso en Slack en el canal “Requisitos Técnicos”.

Estructura del Entregable.

Las Entregas del concurso tendrán tres aspectos importantes a tener en cuenta: La aplicación desplegada y en funcionamiento, el código fuente de todo el proyecto y un documento descriptivo de la solución. A continuación se definen cada uno de estos elementos.

Código Fuente (GitHub)

Los concursantes deben poner a disposición de los evaluadores todo el código fuente desarrollado. Para ello, el equipo de evaluadores de Intellisys Challenge creará repositorios en GitHub para cada uno de los equipos de desarrollo y les dará los respectivos permisos de lectura y escritura. En estos repositorios los equipos deberán compartir el código fuente del proyecto junto al documento de entrega que se describirá más adelante en la sección “*Documentación*”.

El equipo de Intellisys Challenge, para el correr y evaluar los proyectos, asociará la aplicación de cada equipo en Heroku. Heroku es una plataforma online que permite el despliegue de aplicaciones en diferentes lenguajes. El equipo de Intellisys Challenge creará el espacio para el despliegue de cada aplicación asociado al repositorio de GitHub de cada uno. De manera que cada equipo solo debe subir su proyecto y mantenerlo actualizado en el repositorio correspondiente en GitHub.

NOTA Importante:

- Dentro del respectivo repositorio de GitHub los equipos deben ubicar siempre la última versión desplegable de su aplicación en el branch “**master**”. Este branch es desde el que será desplegada la aplicación para ser evaluada. El despliegue se describirá en el próximo acápite “*Aplicación desplegada y en Funcionamiento*”.
- En la raíz de la aplicación deben incluir dos ficheros muy importantes para el despliegue de la aplicación en Heroku:
 - **requirements.txt**: En este fichero se deben poner todas las dependencias necesarias para el proyecto. Si no hiciera falta ninguna, de todas maneras es necesario ubicar este fichero vacío. Ver <https://devcenter.heroku.com/articles/deploying-python#declare-app-dependencies>
 - **procfile**: Configuraciones para la ejecución de la aplicación. Ver <https://devcenter.heroku.com/articles/deploying-python#the-procfile>

EJEMPLO: Como ejemplo se adjunta un fichero comprimido “*Hello_Heroku_Deploy_World.zip*” que tiene una aplicación desplegable de ejemplo con los ficheros y configuraciones básicas anteriores.

Al finalizar cada etapa del concurso, los evaluadores retirarán los permisos de escritura a los concursantes de manera que, luego de la fecha tope, no se podrán hacer actualizaciones ni subidas de nuevos archivos. Estos permisos serán otorgados nuevamente luego de las etapas de corrección e

inicio de nuevas etapas, de manera que ya podrán actualizar y subir nuevamente archivos y elementos del proyecto.

Base de Datos

Las bases de datos deben contener información ya insertada para poder comprobar el funcionamiento del sistema.

Las credenciales (usuario y contraseña) para cada usuario dentro del sistema y sus privilegios se debe describir en el documento de la entrega. Este documento será descrito más adelante en el acápite “*Documentación*”.

App Android

Si el proyecto consta de aplicaciones móviles, en la propia aplicación Web se recomienda ubicar un enlace para descargar la app móvil. De igual manera, el apk debe incluirse entre los ficheros subidos al repositorio junto al código fuente.

Documentación

Los concursantes deberán entregar una documento **PDF** que describa la solución realizada, como funciona la misma y aspectos importantes como los diagramas de clases y de bases de datos. Igualmente debe detallarse la estructura del proyecto y su código fuente, de manera que los evaluadores comprendan fácilmente dónde encontrar cada aspecto que deseen dentro de la estructura de directorios y paquetes y también la información de usuarios con los que puedan loguearse en el sistema.

Este documento deben depositarlo igualmente en la raíz del repositorio de GitHub.

Adjunto a este documento se encuentra otro llamado “**Plantilla de Documento**” que contiene una guía de cómo conformar la entrega final.

Criterios Para Evaluación y Selección

Para la selección del equipo ganador y determinar aquellos equipos que avanzan a la próxima ronda en cada corte se tendrá en cuenta los criterios que a continuación se describen para un mejor entendimiento.

Innovación

Para evaluar la innovación de las propuestas se tendrá en cuenta de forma general la capacidad innovadora y diferenciadora de la propuesta.

Corrección

Capacidad de darle solución de forma correcta a cada uno de los requisitos planteados. Tendrá en cuenta el nivel de integración de diferentes aplicaciones. Se recomienda fuertemente que la aplicación cuente desde sus inicios con los siguientes elementos.

- **Juego de datos iniciales:** Al publicar el sistema debe contar con un juego de datos de manera que exista información para correr el sistema y todas sus funcionalidades sin necesidad de perder tiempo en la entrada de datos por parte de los evaluadores. Importante... Esto no elimina que la solución debe tener la capacidad de poder introducir nuevos datos en caso deseado.
- **Validaciones:** El sistema debe validar correctamente la entrada de datos de manera que no se generen errores del sistema por esta causa y se evite la falta de integridad de la información.

Experiencia de Usuario

Para evaluar la Experiencia de Usuario de las propuestas se tendrá en cuenta de forma general qué tan fácil y amena ha sido la experiencia del usuario al aplicar la solución. Esto incluye entre otros los siguientes aspectos específicos:

- **Usabilidad:** Facilidad y eficiencia en el uso de la aplicación. Los procesos deben implementarse de la manera más sencilla posible y deben ser intuitivos para dar lugar a que se comprenda fácilmente cómo utilizar el sistema.
- **Uso de AJAX:** Para mejorar la experiencia del usuario, disminuir los intervalos de refrescamiento del navegador.

Profesionalidad

Para evaluar Profesionalidad en las propuestas se tendrá en cuenta de forma general la solidez del paquete tecnológico, la arquitectura y el ordenamiento del código de la propuesta seleccionada. Esto incluye aspectos específicos que a continuación se detallarán:

- **Estándar de codificación o estilo de código:** Se evaluará que el código presentado cumpla con las normativas de codificación descritas en los documentos que se encuentran adjuntos a esta guía “Python Enhancement Proposal” (PEP8).

- **Control de Versiones:** Se medirá que cada entregable contenga el código del proyecto correctamente versionado siguiendo las pautas generales recomendadas por GIt en <poner vínculo> y en especial será de carácter obligatorio al menos contar con las siguientes pautas:
 - **Establecer Configuración global:** Cada integrante del equipo tendrá que tener configurado su **username** y **email** que lo identifique y deberá coincidir con los datos suministrados en el momento de inscripción.
 - **Branches (Ramas básicas):** Al menos cada proyecto deberá tener 2 ramas **master** y **develop**. Siendo Master la que será sometida a evaluación en cada entrega.
- **Documentación:** La documentación del proyecto entregada debe cumplir con lo descrito en el acápite “Documentación” .

NOTA: La detección de plagios o copias en las soluciones del concurso serán penalizadas con la descalificación automática del proyecto.

Enlaces Útiles

Para tener una guía para el trabajo y estudio Django, como FrameWork MVC/MVT, ver las siguientes direcciones y libros.

<https://docs.djangoproject.com/en/1.9/faq/general/>

<https://www.youtube.com/watch?v=fICgJXMIvYA>

[*Two Scoops of Django, Best Practices For Django 1.8*](#)

[Django Design Patterns and Best Practices](#)

[Learning Django Web Development](#)