

Django Technical Test

LEVEL: Intermediate

SKILLS: Required: Django, Python, HTML/CSS, Javascript, Bootstrap
Optional: Angular/Rest, Docker

1. Objetivo

El objetivo de éste test práctico consiste en realizar una aplicación simple en el framework Django que simule un administrador de stock. Para ello se proveerán una serie de User Stories simulando los requerimientos, y algunos requerimientos técnicos mínimos que debe respetar la aplicación.

Consigna para presentación del trabajo

1. Crear un proyecto Django 2.0 (mínimo) de acuerdo a los requerimientos técnicos y funcionales utilizando las mejores prácticas del framework, y las consideraciones de diseño que crea convenientes
2. Setear un motor de base de datos a elección según requerimientos técnicos, e incluir las migraciones necesarias para crear las tablas
3. Una vez finalizado, subir el código completo a un repositorio de github.com y enviar el acceso para poder revisarlo, junto con instrucciones particulares si hubiera para instalar la app.
4. Opcional: idealmente, la aplicación podrá ser entregada en un contenedor Docker
5. Opcional: se valorará la inclusión de alguna librería o framework de Javascript (React o Angular) para las funciones que crea convenientes

2. Requerimientos de la aplicación

2.1. REQUERIMIENTOS TECNICOS

1. Versión Django: 2.0 o superior
2. Versión Python: 3
3. Motor de Base de datos: Postgres o MySQL
4. JQuery: 1.9 o superior
5. Framework CSS (*): Bootstrap
6. Framework Angular (*)
7. Librería React (*)

(*) *Opcional*

2.2. ASSUMPTIONS

1. Se asume que al ser una aplicación simple, no se requieren medidas de seguridad como roles/permisos, salvo el login
2. Los usuarios / passwords se crean desde el Admin de Django
3. Las Categorías de productos se crean desde el Admin de Django

2.3. REQUERIMIENTOS FUNCIONALES / USER STORIES

1. Como Usuario, debo poder loguearme al sistema con mi usuario y password

CRITERIOS DE ACEPTACION

- El usuario y password se setean desde el Admin de Django
- Si la password es incorrecta, debe volver a mostrar el form de login con el mensaje de error
- Si el login es correcto, debe mostrar la consola de Stock

2. Como Usuario, debo tener una consola de Stock donde lista todos artículos

CRITERIOS DE ACEPTACION

- La consola muestra los artículos en una tabla con las siguientes propiedades: CODIGO, NOMBRE ITEM, PRECIO UNITARIO, CANTIDAD EN STOCK
- La lista no tiene ordenamiento
- La lista no tiene paginación

NICE TO HAVE

- Se puede realizar una búsqueda por nombre del item (puede ser en javascript)

3. Como Usuario, debo poder agregar un artículo al stock

CRITERIOS DE ACEPTACION

- La acción para agregar se encuentra en la consola de Stock
- El artículo tiene los siguientes datos: CODIGO (*), NOMBRE ITEM (*), TIPO DE ITEM (*) (por ejemplo áridos, mampostería, grifería, etc.), PRECIO UNITARIO, CANTIDAD EN STOCK
- Los tipos de item son customizables por el usuario (a efectos del test los tipos se administran desde el Admin)
- El alta se realiza con un formulario de Django
- Se debe validar que estén todos los campos obligatorios completos para dar el alta
- Una vez que se completó el alta, se debe redireccionar a la Consola

(*) *estos campos son obligatorios*

4. Como Usuario, debo poder eliminar un artículo al stock

CRITERIOS DE ACEPTACION

- La acción para eliminar se encuentra en la consola de Stock (con un botón o ícono al lado de cada artículo listado)
- Se pide confirmación antes de eliminar
- Se muestra confirmación luego de eliminar el artículo

5. Como Usuario, debo poder consultar un artículo del stock

CRITERIOS DE ACEPTACION

- El acceso al artículo se realiza desde la consola de Stock
- Se muestran los datos del artículo seleccionado
- Se muestra una barra de acciones con un botón que permite editar el artículo

6. Como Usuario, debo poder editar un artículo del stock

CRITERIOS DE ACEPTACION

- La edición se realiza mediante un formulario Django
- Se marcan los campos que son obligatorios con algún estilo o indicador (por ej. un “*”)
- Se valida que los campos obligatorios estén completos. Si falta algún campo, se debe informar al usuario cuál es el que falta (con algún estilo en el campo, o mensaje)

7. Como Administrador, debo poder gestionar USUARIOS

CRITERIOS DE ACEPTACION

- La administración se realiza desde el Admin de Django

8. Como Administrador, debo poder gestionar los TIPOS de Items

CRITERIOS DE ACEPTACION

- La administración se realiza desde el Admin de Django