

Entrega Final

- ☐ Link en github.
- ☐ README.md con ubicacion de funcionalidades o pasos a seguir para probar las cosas y los nombres de los integrantes del grupo.
- ☐ Nombre del repo "Entrega1-Apellidos".
- ☐ Video explicativo de 10 min max (subido en el proyecto o cargado el link al mismo).
- ☐ Herencia implementada en los templates.
- ☐ Exista gitignore con:

<carpeta del entorno virtual>

•pycache__

db.sqlite3

media

Estos ultimos son por el hecho de no compartir la info de tu bd y, aparte, las imagenes son archivos muy pesados que no es recomendable tenerlos en el repo. En cambio, las imagenes que sean parte del codigo del proyecto deberian guardarse en la carpeta static.

- ☐ Existencia del archivo requirements.txt actualizado.
- ☐ Rama principal "main".
- ☐ Estructura del los archivos del proyecto:

<nombre de la carpeta del proyecto / repositorio>

| -> <app> accounts

| -> <app> inicio

| -> <app> mensajeria (?)

| -> <carpeta de configuraciones>

- | -> <templates>
- | -> manage.py
- | -> .gitignore
- | -> <carpeta del entorno virtual>
- | -> db.sqlite3
- | -> requirements.txt

- ☐ Las clases deben definirse con PascalCase (la primera letra en mayuscula seguida de las demas en minúscula y, si posee mas de una palabra, cada palabra debe arrancar con mayuscula) y lo que debe ser en snake_case (minusculas y separando las palabras con guion bajo) son las variables, funciones, path names, blocks de los templates, etc. No usar simbolos, caracteres especiales, la ñ para nombres de variables/funciones/etc.
- ☐ Uso de un mismo idioma en todo el proyecto.
- ☐ Borrar todo codigo que no este en uso (aumenta la legibilidad y mejora el rendimiento del proyecto). Ej: imports que no se usen los archivos, codigo comentado sin uso, etc.
- ☐ No dejar botones y accesos que no tengan funcionalidad.
- ☐ Uso de minimo 2 clases basadas en vista.
- ☐ Uso de minimo un mixin en una CBV y un decorador en una view comun.
- ☐ Tener en cuenta al manejar forms con imagenes de adaptar el template, y la vista... no solo el modelo y el formulario.

- ☐ Un home.
- ☐ Agregar un apartado "Acerca de nosotros" en el path "about/". Esta es una simple vista, como las que vimos al principio de django, que contenga una breve reseña suya.
- ☐ 1 clase Post/Blog/Page que debe tener como minimo los campos de : titulo, subtitulo, contenido (este campo implementando el RichText del paquete ckeditor), autor, fecha_creacion e imagen. (a partir de aca lo nombro como "objeto")

- ☐ Acceso a una vista de listado de objetos con info minima de cada uno (path 'pages/) y que contenga un buscador por el atributo titulo.
- ☐ En el listado debería aparecer un cartel que informe que no hay creados objetos hasta el momento (para que el usuario sepa que ahi va a aparecer el listado de los objetos cuando se carguen) o que la búsqueda no encontro ninguno con los datos buscados.
- ☐ Accesos a vistas para poder crear, editar y borrar un objeto. En la creacion y edicion pueden generar la fecha y autor por medio de las vistas y no usando inputs en el formulario.
- ☐ Acceso desde el listado a cada post para ver la info completa del mismo (path 'pages/<id> ').

En esta informacion debera mostrarse como minimo: titulo, subtítulo, contenido con editor de texto, la imagen y la fecha de posteo.

- ☐ Tener una app “accounts” para manejar las vistas relacionadas a los usuarios.
- ☐ Acceso a una pantalla de registro donde se solicite solo nombre de usuario, email y contraseña.
- ☐ Acceso a una pantalla de login.
- ☐ (Logueado) Acceso a una pantalla perfil donde se muestre la informacion del usuario Esta como minimo debe contener: nombre, avatar, descripcion, link a una pagina, email.
- ☐ (Logueado) Acceso a una pantalla para que el usuario pueda modificar su informacion. Esta debe permitir modificar los datos nombrados en el punto anterior y, ademas, el password.
- ☐ Contar con el apartado “admin/” con toda la configuracion requerida para poder manejar la info de la pagina desde este apartado.
- ☐ App de mensajería entre usuarios. (punto extra para llegar al 10)
- ☐ PUNTAZO A TENER EN CUENTA! PROBAR, PROBAR Y PROBAR ANTES DE SUBIR EL CODIGO A GITHUB... (no apurarse a hacer el commit y subir los cambios porque puede generar algun problema sin darnos cuenta)