

Piscis

Platform for Interactive Search and Citizen Science

Introducción

Históricamente, la identificación de características visuales en imágenes resulta fundamental para el estudio de múltiples fenómenos astronómicos como la clasificación morfológica de galaxias, la detección de supernovas/kilonovas, la clasificación espectroscopía de objetos estelares, etc. En los tiempos modernos el crecimiento exponencial en volumen y complejidad de los datos generados en astronomía dificulta considerablemente el análisis manual de los mismos, volviéndolo en muchos casos inviable.

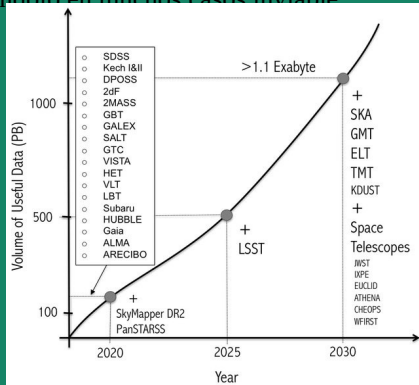


Figura 1. Volumen de datos medido en PB de los relevamientos astronómicos más importantes de los próximos 10 años. Rosa. E (2021)

Objetivo

Con el fin de favorecer la ciencia de datos y fomentar la ciencia ciudadana, construimos una aplicación informática flexible para ser aplicada en problemas de clasificación visual o similares. Esta herramienta se denominó **PISCIS: Platform for Interactive Search and Citizen Science**, y la misma está destinada para facilitar la recolección de datos científicos, generar interés en la ciencia por parte del público general y presentar la información recolectada en una base de datos organizada que sea de fácil lectura y acceso para los usuarios.

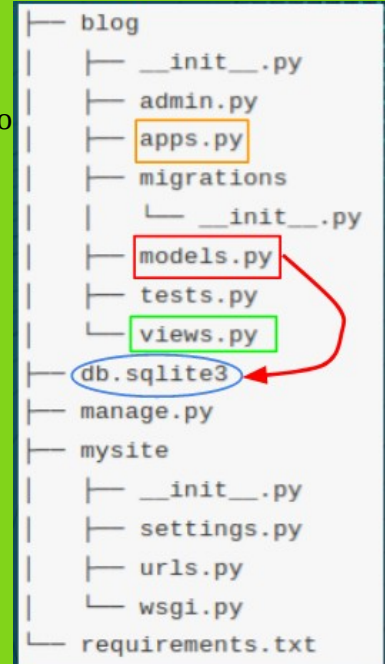
Metodología

A partir de la creación de encuestas concisas acompañadas por imágenes, junto con espacios para agregar información y ejemplos, PISCIS cumple con su **función principal**, la **generación de catálogos de clasificación de imágenes**. Una diferencia crucial frente a aplicaciones preexistentes (como Zooniverse), es que en nuestra plataforma el usuario cuenta con un framework base que **se puede adaptar a su necesidad, en lugar de limitarse a opciones preestablecidas** de cómo realizar su investigación.

Considerando los anteriores requerimientos, desarrollamos PISCIS sobre **Django**, el cual cuenta con un conjunto estandarizado de conceptos, prácticas y criterios para desarrollar páginas web. Este framework web **sigue el patrón de diseño MVT: Modelo-Vista-Template**, y su lenguaje principal de programación es Python.

Django

Estructura básica de Django los **Modelos** definen que objetos se guardan en la **Base de datos** y sus propiedades. Las **Vistas** determinan cuales de esas propiedades se muestran al publico. Y los **Templates** definen cómo se muestran esas propiedades.



Modelos en PISCIS

```
class Images(models.Model):
    picture = models.ImageField(upload_to="image")

class Choice(models.Model):
    usuario = models.ForeignKey(User, on_delete=models.CASCADE, null=False)
    imagen = models.ForeignKey(
        Images, on_delete=models.CASCADE, blank=True, null=True
    )
    CHOICE_TEXT = (("A", "Alta"), ("B", "Baja"), ("C", "Media"))
    voto = models.CharField(max_length=20, choices=CHOICE_TEXT)
```

Images: Define las imágenes a votar en la encuesta.

Choice: Define las opciones de clasificación.

Visualización



Al desplegar PISCIS las encuestas se muestran de esta forma. Recuerden que **las imágenes y opciones son completamente editables**.

Links importantes

Proyecto de Ciencia ciudadana:

¡Piscis ya está en siendo usado para la recolección de datos!

<https://paresdegaxias.iate.conicet.unc.edu.ar/>

Descarga:

<https://github.com/vanedaza/piscis>

Documentación:

<https://piscis.readthedocs.io/en/latest/>