

# WEB APPLICATIONS

**Salas Figueroa Jesus Nahataen**

*Universidad Tecnológica de Tijuana.  
TSU. Tecnologías de la información.  
Entornos virtuales y negocios digitales.  
Docente: DR. Parra Galviz Ray Burnet.  
Email: [0322103855@ut-tijuana.edu.mx](mailto:0322103855@ut-tijuana.edu.mx)*

## I. INTRODUCCIÓN

IN this document I will cover a series of inheritance activities in django

### Django inheritance

In Django, inheritance is a feature that allows you to create new models by extending or inheriting from existing models. This is a powerful concept in object-oriented programming and can be useful in various scenarios when designing a database schema for your web application.

Django's inheritance features provide a way to reuse and extend models efficiently, ensuring that you can create a database schema that reflects the relationships and properties of your application's data in a structured and organized manner. [1]

There are two main types of inheritance in Django models:

**Abstract Base Classes:** Abstract base classes are used when you want to define common fields and methods that should be shared by multiple models. These abstract classes are not meant to be instantiated on their own. Instead, they serve as templates for other models to inherit from. You can create an abstract base class by setting the abstract option to True in the model's Meta class. [1]

**Multi-table Inheritance:** In multi-table inheritance, each model in the inheritance hierarchy corresponds to a separate database table. The child models inherit the fields from the parent model and can also add new fields or override existing ones. This allows you to store each model in a separate table and perform queries on them independently. [1]

### conclusion

In conclusion, Django's inheritance feature allows you to create new models by extending existing ones, enabling efficient reuse and extension of model fields and methods. There are two main types of inheritance in Django models: Abstract Base Classes, which define common fields and methods for multiple models, and Multi-table Inheritance, where each model corresponds to a separate database table, inheriting and extending fields from a parent model. This flexibility helps in creating a well-structured and organized database schema for your web application. [1]

## REFERENCES

- [1] Django, "Django project," [Online]. Available: Django. (s. f.). Django Project. Recuperado 5 de noviembre de 2023, de <https://docs.djangoproject.com/en/4.2/topics/templates/>. [Accessed 05 11 2023].