

Modules

[models.EmpleadoDTO](#)

[database.dbconnection](#)

Functions

```
add_employee(entry_nombre: str, entry_seg_social: str, entry_pagas_extras: str, entry_email: str, entry_irpf: str, entry_salario: str, entry_telefono: str, entry_puesto: str, entry_departamento: str, entry_genero: str, entry_afiliacion: str, entry_datos_bancarios: str, entry_fecha_nacimiento: str, entry_nif: str, entry_direccion: str, entry_fecha_inicio: str)
    Hace una validación de los datos y los inserta en la base de datos. Si es una cadena vacía, asigna valores por defecto.

obtener_empleado()
    Obtiene los datos del último empleado insertado en la base de datos y lo transforma en DTO.

Returns:
    Optional[EmpleadoDTO]: Objeto DTO del empleado si existe, de lo contrario None.

Ejemplo de uso:
>>> from models import EmpleadoDTO

>>> datos = (1, "Juan Pérez", "123456", 2, "juan@example.com", 15.0, 3000.0, "654321098", "Desarrollador", "IT", "M", "Sindicato X", "Cuenta XYZ", "1985-06-15", "12345678A", "Calle Falsa 123", "2020-01-10")
>>> db.obtener_datos = lambda: datos
>>> empleado = obtener\_empleado()

>>> isinstance(empleado, EmpleadoDTO.EmpleadoDTO) or empleado is None
True

>>> empleado.nombre
'Juan Pérez'

>>> empleado.salario
3000.0

>>> isinstance(empleado.id, int)
True

>>> isinstance(empleado.pagas_extras, str)
False

>>> isinstance(empleado.fecha_nacimiento, str)
False
```

Data

Optional = typing.Optional
Tuple = typing.Tuple