**Tutorial 03-B to do in class – Remember to upload the repo link to Teams**

**DIP**

**Antes de iniciar:**

* Terminar los talleres no calificables anteriores
* Este taller muestra el ejemplo de una inversión de dependencias y de una inyección de dependencias.
* ¿Puede identificar donde está la inversión, y donde la inyección?
* Al final vemos un ejemplo de cómo sería sin inversión de dependencias.

1. **An interface**

**Interface**

* Go to *pages* create a file *interfaces.py* with the next content:

|  |
| --- |
| **Add Entire Code** |

from abc import ABC, abstractmethod

from django.http import HttpRequest

class ImageStorage(ABC):

@abstractmethod # any class that inherits from this one must implement this method

def store(self, request: HttpRequest):

pass

1. **A util library**

**Image local storage**

* Go to *pages* create a file *utils.py* with the next content:

|  |
| --- |
| **Add Entire Code** |

from django.core.files.storage import default\_storage

from django.http import HttpRequest

from .interfaces import ImageStorage

class ImageLocalStorage(ImageStorage):

def store(self, request: HttpRequest):

profile\_image = request.FILES.get('profile\_image', None)

if profile\_image:

# Store the image

file\_name = default\_storage.save('uploaded\_images/' + profile\_image.name, profile\_image)

return default\_storage.url(file\_name)

* Go to *helloworld\_project/setting.py*  and add the following content:

**Add bold Code**

#top of the file

from pathlib import Path

**import os**

…

**MEDIA\_URL = '/media/'**

**MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')**

* Go to *helloworld\_project* and create a folder named *media.*
* Go to *helloworld\_project/urls.py*  and add the following content:
* **Add bold Code**

from django.contrib import admin

from django.urls import path, include

**from django.conf import settings**

**from django.conf.urls.static import static**

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('pages.urls')),

path('accounts/', include('accounts.urls')),

]

**if settings.DEBUG:**

**urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)**

1. **A service provider**

**Provider**

* Go to *helloworld\_project/settings.py/* an add the following content at the bottom of the file:

|  |
| --- |
| **Add Entire Code** |

IMAGE\_STORAGE\_CLASS = 'pages.utils.ImageLocalStorage'}

**Registering the new provider**

* In *pages/apps.py*, make the following changes in **bold**.

|  |
| --- |
| **Modify Bold Code** |

from django.apps import AppConfig

**from django.utils.module\_loading import import\_string**

**from django.conf import settings**

class PagesConfig(AppConfig):

default\_auto\_field = 'django.db.models.BigAutoField'

name = 'pages'

**def ready(self):**

**ImageStorageClass = import\_string(settings.IMAGE\_STORAGE\_CLASS)**

1. **Rest of the code**

**View/Controller**

* Go to *pages/views.py* and add the next content:

|  |
| --- |
| **Add bold Code** |

…

from .models import Product

…

**def ImageViewFactory(image\_storage):**

**class ImageView(View):**

**template\_name = 'images/index.html'**

**def get(self, request):**

**image\_url = request.session.get('image\_url', '')**

**return render(request, self.template\_name, {'image\_url': image\_url})**

**def post(self, request):**

**image\_url = image\_storage.store(request)**

**request.session['image\_url'] = image\_url**

**return redirect('image\_index')**

**return ImageView**

**Routes**

* Go to *pages/urls.py* and add these new routes by adding the next lines at the end of the urlspatterns (check **bold**).

|  |
| --- |
| **Modify Bold Code** |

…

**path('image/', ImageViewFactory(ImageLocalStorage()).as\_view(), name='image\_index'),**

**path('image/save', ImageViewFactory(ImageLocalStorage()).as\_view(), name='image\_save'),**

**Template/view**

* Go to *pages/templates* create a folder *images*.
* Go to *pages/templates/image/* create a file *index.html* with the next content:

|  |
| --- |
| **Add Entire Code** |

{% extends 'pages/base.html' %}

{% block title %} Image Storage - DI {% endblock %}

{% block content %}

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">Upload image</div>

<div class="card-body">

<form action="{% url 'image\_save'}" method="post" enctype="multipart/form-data">

{% csrf\_token %}

<div class="form-group">

<label>Image:</label>

<input type="file" name="profile\_image" />

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

<img src="{{ image\_url }}" />

</div>

</div>

</div>

</div>

</div>

{% endblock %}

**Artisan**

* Run the next command in Terminal

|  |
| --- |
| **Execute in Terminal** |

python manage.py runserver

**Execution**

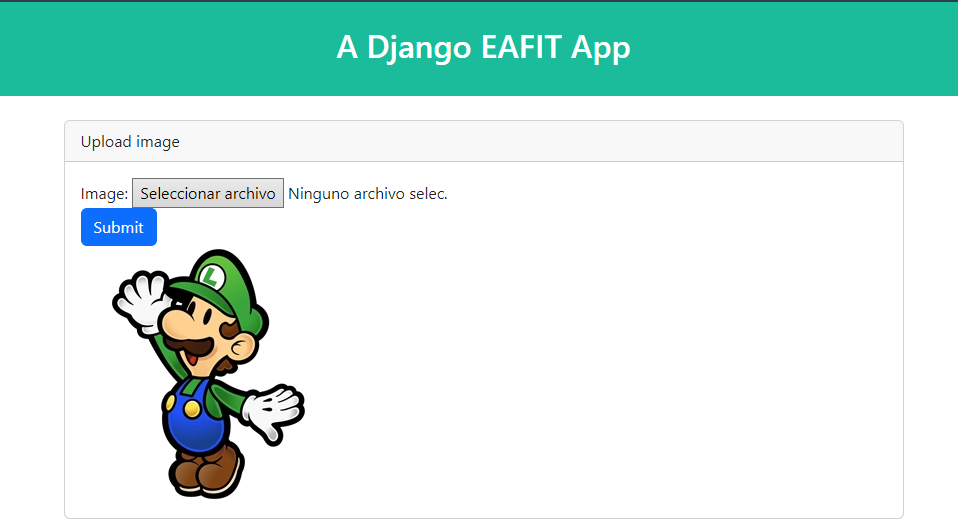
* Go to <http://127.0.0.1:8000/image/>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Upload image**

* Upload a new image



**SAME APPLICATION WITHOUT DEPENDENCY INVERSION**

**Controller**

* Go to *pages/views.py* create a View *ImageNotDIView* with the next content:

|  |
| --- |
| **Add Entire Code** |

class ImageViewNoDI(View):

template\_name = 'images/index.html'

def get(self, request):

image\_url = request.session.get('image\_url', '')

return render(request, self.template\_name, {'image\_url': image\_url})

def post(self, request):

image\_storage = ImageLocalStorage()

image\_url = image\_storage.store(request)

request.session['image\_url'] = image\_url

return redirect('image\_index')}

**Routes**

* Go to *pages/urls.py*and add these new routes by adding the next lines at the end of the file (check **bold**).

|  |
| --- |
| **Modify Bold Code** |

…

**path('imagenotdi/', ImageViewNoDI.as\_view(), name='imagenodi\_index'),**

**path('image/save', ImageViewNoDI.as\_view().as\_view(), name='imagenodi\_save'),**

**View**

* Go to *pages/templates/* create a folder *imagesnotdi*.
* Go to *pages/templates /imagesnotdi/* create a file *index.html* with the next content:

|  |
| --- |
| **Add Entire Code** |

{% extends 'pages/base.html' %}

{% block title %} Image Storage - No DI {% endblock %}

{% block content %}

<div class="container">

<div class="row justify-content-center">

<div class="col-md-8">

<div class="card">

<div class="card-header">Upload image</div>

<div class="card-body">

<form action="{% url 'imagenotdi\_save' %}" method="post" enctype="multipart/form-data">

{% csrf\_token %}

<div class="form-group">

<label>Image:</label>

<input type="file" name="profile\_image" />

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

<img src="{{ image\_url }}" />

</div>

</div>

</div>

</div>

</div>

{% endblock %}

**Execution**

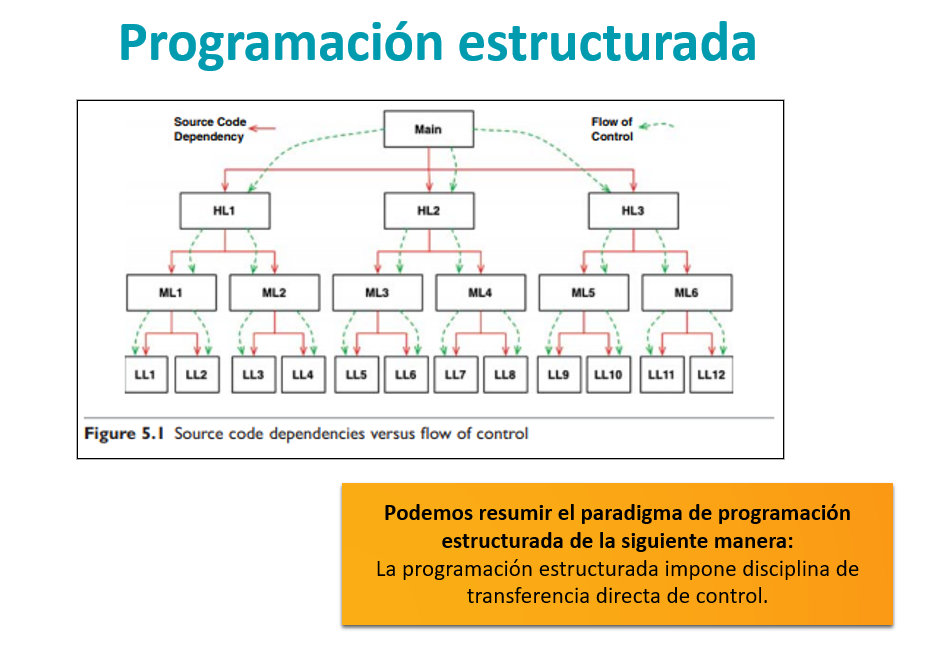
* Go to <http://127.0.0.1:8000/image-not-di>

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Preguntas**

* ¿Puedes entender las diferencias entre las dos propuestas?
* ¿Ventajas/desventajas de cada una?
* Qué tal si tratas de comparar estas propuestas con lo siguiente (ver siguiente página):



Diagrama

Descripción generada automáticamente

**¿Será que con el ejemplo ya te queda clara esa carreta? ¿Sí? ¿No? La próxima clase conversaremos sobre esto**