

**Instituto Tecnológico de Costa Rica**

**Introducción al desarrollo de páginas web**

**Enunciado taller Django**

**Prof. Ericka Solano Fernández**

**Integrantes:**

Andrés Sánchez Gómez - 2017239278

Diego Velásquez Castro - 2017107723

María Venegas Berrocal - 2017097489

Rodrigo Venegas Vega - 2017047627

**Verano 2019 - 2020**

<b>Objetivo</b>	<b>3</b>
<b>Problema a resolver</b>	<b>3</b>
<b>Instrucciones del taller</b>	<b>3</b>

## Objetivo

El principal objetivo de este taller es enseñar brevemente un poco de Django a nuestros compañeros, que con este taller puedan iniciarse en Django y empezar a desarrollar aplicaciones WEB pequeñas. Además, se planea enseñar a desarrollar una aplicación que ayude a los estudiantes a organizar sus tareas.

## Problema a resolver

Se debe crear una lista de tareas por realizar, en la cual se permite añadir, eliminar y editar una tarea, marcarla como realizada y desmarcarla también, desde la interfaz web implementada mediante Django.

## Instrucciones previas al taller

Los pasos previos para llevar a cabo el ejercicio son:

1. Instalar Python y Django (Utilizar el instructivo proporcionado)
2. Descargar el repositorio del ejercicio con los archivos base  
<https://github.com/Endlaze/todo>

## Instrucciones:

1. Dentro de todo\ ejecutar el comando  
**python manage.py runserver**  
para levantar el servidor
2. Seguidamente debe entrar a localhost:8000/ en su navegador, en la cual le aparecerá una página comunicando que la instalación fue exitosa.
3. Crear el archivo **models.py** en **todo-master\todo\_list\** y copiar el siguiente código:

```
from django.db import models

class List(models.Model):
    item = models.CharField(max_length=200)
```

```

completed = models.BooleanField(default=False)

def __str__(self):
    return self.item + ' | ' + str(self.completed)

```

Con esto lo que hacemos es dar el formato de cada tarea (item) en nuestra ToDo-List, dichas tareas tendrán un valor por defecto de **False** (completed).

4. Crear el archivo **forms.py** en **todo-master\todo\_list\** y copiar el siguiente código:

```

from django import forms
from .models import List

class ListForm(forms.ModelForm):
    class Meta:
        model = List
        fields = ["item", "completed"]

```

Estamos creando el from que contendrá los valores de nuestras tareas.

5. Posteriormente, se crea un archivo **views.py** (**todo-master\todo\_list\**) en el cual se crearán las funciones de delete, editar, marcar, desmarcar y agregar. Para ello, debe primero copiar el siguiente código:

```

from django.shortcuts import render, redirect
from .models import List
from .forms import ListForm
from django.contrib import messages

def home(request):
    if request.method == 'POST':
        form = ListForm(request.POST or None)

        if form.is_valid():
            form.save()
            all_items = List.objects.all
            messages.success(request, ('Tarea ha sido agregada a la lista!'))

            return render(request,
"home.html",{ 'all_items':all_items})

```

```

    else:
        all_items = List.objects.all
        return render(request, 'home.html', {'all_items':
all_items})

```

Este código se realiza para recibir la solicitud del usuario de añadir una nueva tarea, por lo que se revisa que los campos de la tarea sean válidos, se guarda y se devuelve un mensaje de éxito, desplegando la nueva tarea en la lista de pendientes del usuario.

#### Delete:

```

def delete(request, list_id):
    item = List.objects.get(pk=list_id)
    item.delete()
    messages.success(request, ('Item Has Been Deleted!'))
    return redirect('home')

```

Busca un ítem con el id a borrar y se extrae del form, envía un mensaje de éxito y redirecciona a home

#### Cross-Off:

```

def cross_off(request, list_id):
    item = List.objects.get(pk=list_id)
    item.completed = True
    item.save()
    return redirect('home')

```

Busca un ítem con el id a marcar para actualizar su campo 'completed' como True, y se guarda.

#### Uncross:

```

def uncross(request, list_id):
    item = List.objects.get(pk=list_id)
    item.completed = False
    item.save()
    return redirect('home')

```

Busca un ítem con el id a desmarcar para actualizar su campo 'completed' como False, y se guarda.

#### Edit:

```

def edit(request, list_id):
    if request.method == 'POST':
        item = List.objects.get(pk=list_id)
        form = ListForm(request.POST or None, instance=item)

```

```

    if form.is_valid():
        form.save()
        messages.success(request, ('Tarea ha sido editada!'))
        return redirect('home')
    else:
        item = List.objects.get(pk=list_id)
        return render(request, 'edit.html', {'item':item})

```

Se obtiene el id del ítem a editar, y revisa si el nuevo campo es válido para guardarlo. Posteriormente muestra un mensaje de éxito comunicando que la tarea ha sido editada mientras redirecciona el usuario a home.

6. Finalmente, debe crear un archivo **urls.py** ( **todo-master\todo\_list\**) y copiar el siguiente código:

```

from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name="home"),
    path('delete/<list_id>', views.delete, name="delete"),
    path('cross_off/<list_id>', views.cross_off, name="cross_off"),
    path('uncross/<list_id>', views.uncross, name="uncross"),
    path('edit/<list_id>', views.edit, name="edit"),
]

```

Aquí se especifica la dirección de la función a ejecutar según la solicitud del usuario.

Finalmente, se debe entrar al archivo **urls.py** localizado en la carpeta **todo-master/todo/urls.py** y descomentar las líneas 18 y 22

```

16  from django.contrib import admin
17  from django.urls import path, include
18  # from todo_list import views
19
20  urlpatterns = [
21      path('admin/', admin.site.urls),
22      # path('', include('todo_list.urls')),
23  ]

```

Una vez realizados estos pasos, deberá actualizar la página y probar las funcionalidades de la lista de quehaceres.

# Resultados

To-Do List

Add Item

Add To List

test!	Cross Off	Delete
new item	Cross Off	Delete
Try again	Cross Off	Delete
Try again!	Cross Off	Delete
The Best Item!	Cross Off	Delete
The Last Item!	Cross Off	Delete

Item Has Been Added To List!

x

test!	Cross Off	Delete
<del>Try again</del>	Uncross	Delete
Try again!	Cross Off	Delete
<del>The Best Item!</del>	Uncross	Delete
The Last Item!	Cross Off	Delete
New Item!	Cross Off	Delete

The Older Item

x

Edit Item

Item Has Been Edited!

x

The Older Item	Cross Off	Delete
----------------	-----------	--------