

## VISUALIZANDO INFORMACIÓN DE BD EN EL PROYECTO

- Dentro del archivo html **principal.html** que se encuentra en al **app inicio**, vamos a agregar una tabla donde visualizaremos nuestros registros.

```
{%block contenido%}
<!-- Tabla de Alumnos -->


| Foto | Matricula | Nombre | Carrera | Turno |
|------|-----------|--------|---------|-------|
|------|-----------|--------|---------|-------|


```

PRINCIPAL.HTML

```
<tbody>
  <tr>
    <td class="text-center">
      <img src="" alt="" class="img-media">
    </td>
    <td class="text-semibold">UTM1234TIC</td>
    <td class="muted">Nathalia</td>
    <td class="muted">Tic</td>
    <td class="muted">Matutino</td>
  </tr>

</tbody>
</table>
<!-- /Tabla de alumnos --&gt;
{%endblock%}</pre>
```

PRINCIPAL.HTML

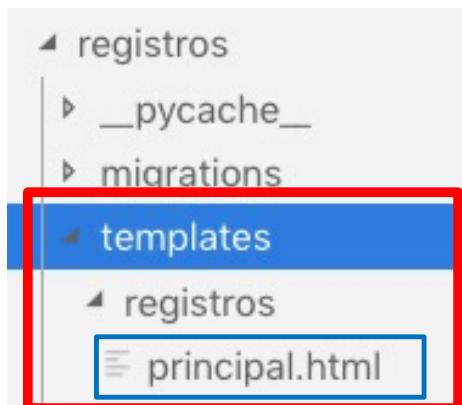
A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:8000`. The page title is "Alumnos" with the subtitle "Lista de Alumnos DSM". On the left, there is a sidebar with a house icon and the text "Principal DSM". Below the sidebar, there are links for "Principal", "Registro", and "Contacto". The main content area displays a table with student information. A red box highlights the first row of the table. The table has columns for "Foto", "Matrícula", "Nombre", "Carrera", and "Turno". The data in the highlighted row is: Matrícula UTM1234TIC, Nombre Nathalia, Carrera Tic, Turno Matutino.

Foto	Matrícula	Nombre	Carrera	Turno
	UTM1234TIC	Nathalia	Tic	Matutino

**RECUERDA QUE POR EL MOMENTO SON DATOS FIJOS.**

## RECUPERANDO DATOS DE LA BD

- Para proceder con la visualización de los alumnos en la tabla del archivo principal.htm y para mantener las funcionalidades organizadas, pasaremos nuestro archivo principal.html a nuestra nueva app registros
- Creamos para la appp registros las carpetas para registrar lor archivos html (**templates/registros**)
- Copiamos el archivo de principal.html de inicio en registros como se observa en la imagen



## RECUPERANDO DATOS DE LA BD

### registros/views.py

La lista de alumnos se mostrará en el template **principal.html** a través de su view **registros**, vamos a visualizar los datos ahí. Recordemos que ahora el archivo que utilizaremos es principal.html de la app registros.

```
def registros(request):
    return render(request, "registros/principal.html")
#Indicamos el lugar donde se renderizará el resultado de
esta vista
```

# urls.py

## ■ DEFINIMOS LA NUEVA RUTA

```
from django.contrib import admin
from django.urls import path
from inicio import views
from django.conf import settings
from registros import views as views_registros
#Importamos la nueva vista de app registros para
poder asignar las rutas de acceso a sus vistas
```

# urls.py

## ■ DEFINIMOS LA NUEVA RUTA

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('',views_registros.registros, name="Principal"),
    #Indicamos que ahora la ruta de principal.html se
    #encuentra en la view de registros
    path('contacto/',views.contacto, name="Contacto"),
    path('registro/',views.formulario, name="Registrar"),
    path('ejemplo/',views.ejemplo, name="Inicio"),
]
```

# registros/views.py

## ■ RECUPERANDO DATOS DE LA BD

```
from django.shortcuts import render
from .models import Alumnos
#Accedemos al modelo Alumnos que contiene la estructura de la tabla.

# Create your views here.

def registros(request):
    alumnos=Alumnos.objects.all()
#all recupera todos los objetos del modelo (registros de la tabla alumnos)

    return render(request,"registros/principal.html",{'alumnos':alumnos})
#Indicamos el lugar donde se renderizará el resultado de esta vista
# y enviamos la lista de alumnos recuperados
```

## registros/principal.html

- Cargando los datos en la tabla

```
<table class="table table-bordered table-striped">
  <thead>
    <tr>
      <th>Foto</th>
      <th>Matricula</th>
      <th>Nombre</th>
      <th>Carrera</th>
      <th>Turno</th>
    </tr>
  </thead>
```

Nos colocamos en la estructura de la tabla

## registros/principal.html

- Cargando los datos en la tabla

El ciclo se repite por cada registro (alumno en la tabla) y para cada uno se imprime una nueva fila

```
<tbody>
  {%
    for alumno in alumnos %
  <tr>
    <td class="text-center">
      
    </td>
    <td class="text-semibold">{{alumno.matricula}}</td>
    <td class="muted">{{alumno.nombre}}</td>
    <td class="muted">{{alumno.carrera}}</td>
    <td class="muted">{{alumno.turno}}</td>
  </tr>
  {%
    endfor %
  </tbody>
```

Las instrucciones {{alumno.xxxxx}} permiten acceder a la columna deseada y muestra el resultado

# Introducción a la programación



- Accedemos al navegador y actualizamos la página principal.
- Debemos de visualizar a los usuario registrados.

Foto	Matricula	Nombre	Carrera	Turno
	UTM1234TIC	Juan	TIC	Matutino

## PERSONALIZANDO EL ADMINISTRADOR

### registros/admin.py

- Daremos formato a la tabla de alumnos separando en columnas:

```
class AdministrarModelo(admin.ModelAdmin):  
    readonly_fields = ('created', 'updated')  
    list_display = ('matricula', 'nombre', 'carrera', 'turno')
```

```
admin.site.register(Alumnos, AdministrarModelo)
```

<input type="checkbox"/>	MATRÍCULA	NOMBRE	CARRERA	TURNO
<input type="checkbox"/>	UTM1234TIC	Juan	TIC	Matutino

# PERSONALIZANDO EL ADMINISTRADOR

## registros/admin.py

- Agregando un formulario de búsqueda:

```
class AdministrarModelo(admin.ModelAdmin):  
    readonly_fields = ('created', 'updated')  
    list_display = ('matricula', 'nombre', 'carrera', 'turno')  
    search_fields = ('matricula', 'nombre', 'carrera', 'turno')
```

The screenshot shows the Django Admin interface for a 'AdministrarModelo' model. At the top, there is a search bar consisting of a magnifying glass icon, a text input field, and a 'Buscar' button. This search bar is highlighted with a thick red border. Below the search bar, the page header displays the date '·2021 25 de junio'. Underneath the header, there is a toolbar with an 'Acción:' dropdown menu set to '-----', a 'Ir' button, and a status message '0 de 1 seleccionado'. The main content area contains a table with four columns: 'MATRÍCULA', 'NOMBRE', 'CARRERA', and 'TURNO'. A single row is listed, showing 'UTM1234TIC' in the 'MATRÍCULA' column, 'Juan' in the 'NOMBRE' column, 'TIC' in the 'CARRERA' column, and 'Matutino' in the 'TURNO' column. Each column header has a corresponding checkbox to its left.

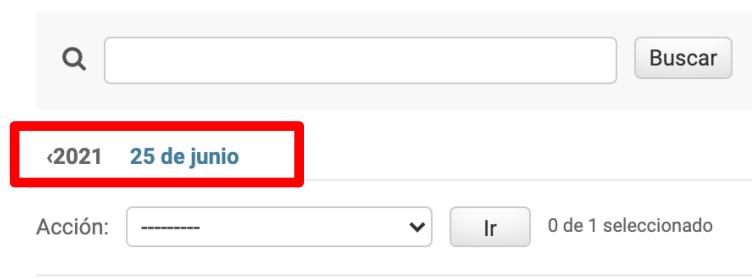
<input type="checkbox"/>	MATRÍCULA	NOMBRE	CARRERA	TURNO
<input type="checkbox"/>	UTM1234TIC	Juan	TIC	Matutino

## PERSONALIZANDO EL ADMINISTRADOR

### registros/models..py

- Agregando búsqueda por fecha

```
class AdministrarModelo(admin.ModelAdmin):  
    readonly_fields = ('created', 'updated')  
    list_display = ('matricula', 'nombre', 'carrera', 'turno')  
    search_fields = ('matricula', 'nombre', 'carrera', 'turno')  
    date_hierarchy = 'created'
```



# PERSONALIZANDO EL ADMINISTRADOR

## registros/admin.py

### ■ Agregando filtro lateral

```
class AdministrarModelo(admin.ModelAdmin):
    readonly_fields = ('created', 'updated')
    list_display = ('matricula', 'nombre', 'carrera','turno')
    search_fields = ('matricula','nombre','carrera','turno')
    date_hierarchy = 'created'
    list_filter = ('carrera','turno')
```

### FILTRAR

Borrar todos los filtros

Por carrera

Todas

TIC

Por turno

Todas

Matutino

## PRÁCTICA – ADMIN DJANGO

- Realizar lo siguiente para el proyecto de CursosDjango
  - Crear una nueva app cursos
  - Crear un modelo para administrar los cursos almacenando los datos que consideres necesarios usa almenos 5 tipos de datos diferentes,(debes incluir una imagen y fecha de creación).
  - Dar formato al panel de administrador para que muestre:
    - Como nombre de la APP → CONVOCATORIAS
    - Como nombre de la lista → Cursos
    - Ordenar los datos por fecha de creación de más antigüa a más reciente
    - Modificar los nombres de los campos que se visualizan en el formulario de registro del administrador.
    - Visualiza los campos automáticos.
  - Personaliza el administrador para:
    - Dar formato a la tabla donde se visualizan los cursos registrados (elige los campos que deseas mostrar).
    - Agrega una barra de búsqueda.
    - Agrega una búsqueda por fecha.
    - Agrega una barra de filtro lateral (elige los campos que deseas filtrar).