

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 -->  
<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>
```

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 -->
{%endblock%}
```

PRINCIPAL.HTML



Principal
DSM

Principal

Registro

Contacto

Alumnos

Lista de Alumnos DSM

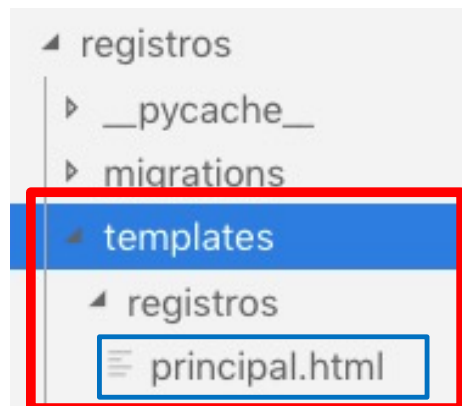
Principal

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 app registros las carpetas para registrar los 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

- 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')
```



A screenshot of the Django admin interface. At the top, there is a search bar with a magnifying glass icon on the left and a 'Buscar' button on the right. The search bar is highlighted with a red rectangular box. Below the search bar, there is a breadcrumb trail showing '<2021' and '25 de junio'. Below that, there is an 'Acción:' label followed by a dropdown menu with a downward arrow, an 'Ir' button, and the text '0 de 1 seleccionado'. Below this, there is a table with four columns: 'MATRÍCULA', 'NOMBRE', 'CARRERA', and 'TURNO'. The first row of the table has a checkbox, the value 'UTM1234TIC', the name 'Juan', the career 'TIC', and the shift 'Matutino'.

<2021 25 de junio

Acción: ----- Ir 0 de 1 seleccionado

<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'
```

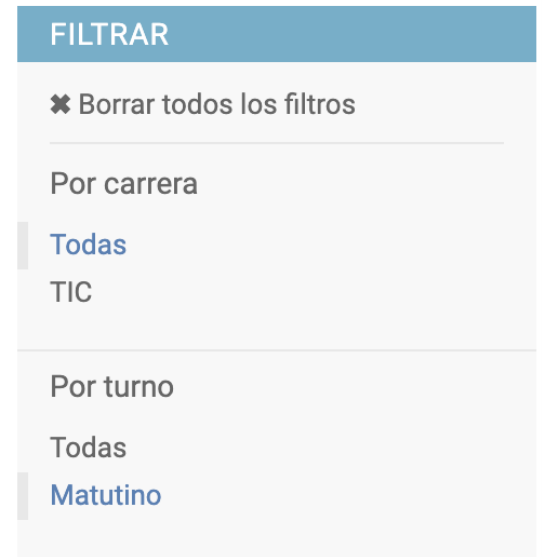
The screenshot shows a portion of the Django Admin interface. At the top, there is a search bar with a magnifying glass icon and a 'Buscar' button. Below the search bar, there is a date filter section. The date filter is highlighted with a red rectangle and contains the text '<2021' and '25 de junio'. Below the date filter, there is an 'Acción:' label, a dropdown menu with a downward arrow, and a 'Ir' button. To the right of the 'Ir' button, it says '0 de 1 seleccionado'.

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')
```



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 al menos 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 antigua 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 desees mostrar).
 - Agrega una barra de búsqueda.
 - Agrega una búsqueda por fecha.
 - Agrega una barra de filtro lateral (elige los campos que desees filtrar).