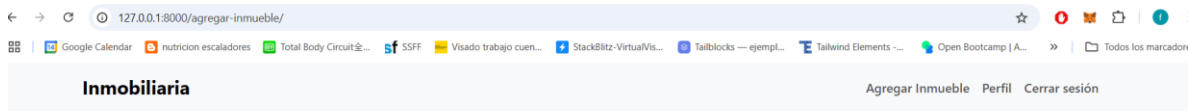


Requerimientos

1. Crear página web básica donde arrendadores puedan agregar nuevos inmuebles.



- a. Generar las rutas para la vista para agregar nuevas viviendas.

```
gestion_inmuebles > urls.py > ...
1  from django.urls import path
2  from django.contrib.auth.views import LoginView, LogoutView
3  from . import views
4
5  urlpatterns = [
6      path('register/', views.register, name='register'), # URL para el registro
7      path('login/', LoginView.as_view(template_name='login.html'), name='login'), # URL para iniciar sesión
8      path('logout/', LogoutView.as_view(template_name='logout.html'), name='logout'),
9      path('', views.home, name='home'),
10     path('perfil/', views.perfil, name='perfil'),
11     path('agregar-inmueble/', views.agregar_inmueble, name='agregar_inmueble'),
12     path('inmueble-exitoso/', views.inmueble_exitoso, name='inmueble_exitoso'),
13 ]
14
```

- b. Generar el objeto de formulario.

```
class InmuebleForm(forms.ModelForm):
    class Meta:
        model = Inmueble
        fields = ['nombre', 'descripcion', 'm2_construidos', 'm2_totales', 'estacionamientos',
                  'habitaciones', 'banos', 'direccion', 'comuna', 'tipo_inmueble', 'precio_mensual']
        widgets = {
            'descripcion': forms.Textarea(attrs={'rows': 3}),
        }
```

- c. Agregar la función para guardar el objeto.

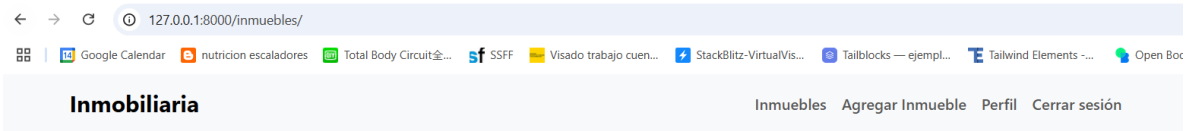
```
def agregar_inmueble(request):
    if request.method == 'POST':
        form = InmuebleForm(request.POST)
        if form.is_valid():
            # Guarda el inmueble y asocia al arrendador actual
            inmueble = form.save(commit=False)
            inmueble.arrendador = request.user.arrendador # Asegura que el arrendador esté asociado
            inmueble.save()
            return redirect('inmueble_exitoso') # Redirige a una página de éxito, puedes crearla
        else:
            form = InmuebleForm()

    return render(request, 'agregar_inmueble.html', {'form': form})

def inmueble_exitoso(request):
    return render(request, 'inmueble_exitoso.html')
```

2. Crear página web básica donde arrendadores puedan actualizar/borrar un inmueble existente.

a. Generar las rutas para la vista para actualizar las viviendas por usuario.



Lista de Inmuebles		
<p>Casa Cordillera</p> <p>Tipo: Casa</p> <p>Descripción: Casa en la cordillera</p> <p>Dirección: combarbala 0906</p> <p>Precio mensual: \$400000.00</p> <p>Editar Eliminar</p>	<p>Casa Cordillera</p> <p>Tipo: Casa</p> <p>Descripción: Casa en la cordillera</p> <p>Dirección: combarbala 0906</p> <p>Precio mensual: \$300000.00</p> <p>Editar Eliminar</p>	<p>Casa Costa</p> <p>Tipo: Casa</p> <p>Descripción: playa</p> <p>Dirección: Ohiggins 88</p> <p>Precio mensual: \$300000.00</p> <p>Editar Eliminar</p>

```

from django.urls import path
from django.contrib.auth.views import LoginView, LogoutView
from . import views

urlpatterns = [
    path('register/', views.register, name='register'), # URL para el registro
    path('login/', LoginView.as_view(template_name='login.html'), name='login'), # URL para iniciar sesión
    path('logout/', LogoutView.as_view(template_name='logout.html'), name='logout'),
    path('', views.home, name='home'),
    path('perfil/', views.perfil, name='perfil'),
    path('agregar-inmueble/', views.agregar_inmueble, name='agregar_inmueble'),
    path('inmueble-exitoso/', views.inmueble_exitoso, name='inmueble_exitoso'),
    path('inmueble/<int:pk>/editar/', views.editar_inmueble, name='editar_inmueble'),
    path('inmueble/<int:pk>/eliminar/', views.eliminar_inmueble, name='eliminar_inmueble'),
    path('inmuebles/', views.lista_inmuebles, name='lista_inmuebles'),
]

```

b. Generar el objeto de formulario en base a él modelo definido.

```

class InmuebleForm(forms.ModelForm):
    class Meta:
        model = Inmueble
        fields = [
            'nombre', 'descripcion', 'm2_construidos', 'm2_totales', 'estacionamientos',
            'habitaciones', 'banos', 'direccion', 'comuna', 'tipo_inmueble', 'precio_mensual'
        ]
        widgets = {
            'descripcion': forms.Textarea(attrs={'rows': 3, 'class': 'form-control'}),
            'nombre': forms.TextInput(attrs={'class': 'form-control'}),
            'm2_construidos': forms.NumberInput(attrs={'class': 'form-control'}),
            'm2_totales': forms.NumberInput(attrs={'class': 'form-control'}),
            'estacionamientos': forms.NumberInput(attrs={'class': 'form-control'}),
            'habitaciones': forms.NumberInput(attrs={'class': 'form-control'}),
            'banos': forms.NumberInput(attrs={'class': 'form-control'}),
            'direccion': forms.TextInput(attrs={'class': 'form-control'}),
            'comuna': forms.Select(attrs={'class': 'form-control'}),
            'tipo_inmueble': forms.Select(attrs={'class': 'form-control'}),
            'precio_mensual': forms.NumberInput(attrs={'class': 'form-control'}),
        ]

```

c. Agregar la función para actualizar el objeto..

- a. Generar las rutas para ver las viviendas.

```
from django.urls import path
from django.contrib.auth.views import LoginView, LogoutView
from . import views

urlpatterns = [
    path('register/', views.register, name='register'), # URL para el registro
    path('login/', LoginView.as_view(template_name='login.html'), name='login'), # URL para iniciar sesión
    path('logout/', LogoutView.as_view(template_name='logout.html'), name='logout'),
    path('', views.home, name='home'),
    path('perfil/', views.perfil, name='perfil'),
    path('agregar-inmueble/', views.agregar_inmueble, name='agregar_inmueble'),
    path('inmueble-exitoso/', views.inmueble_exitoso, name='inmueble_exitoso'),
    path('inmueble/<int:pk>/editar/', views.editar_inmueble, name='editar_inmueble'),
    path('inmueble/<int:pk>/eliminar/', views.eliminar_inmueble, name='eliminar_inmueble'),
    path('inmuebles/', views.lista_inmuebles, name='lista_inmuebles'),
    path('disponibles/', views.lista_disponibles, name='lista_disponibles'),
]
```

- b. Crear la vista y el controlador que le permitan enlistar las viviendas.

```
@login_required
def lista_disponibles(request):
    # Filtrar inmuebles disponibles
    inmuebles_disponibles = Inmueble.objects.filter(arrendador__isnull=False) # Ajusta el filtro según tu modelo
    return render(request, 'lista_disponibles.html', {'inmuebles': inmuebles_disponibles})
```

(2 Puntos)