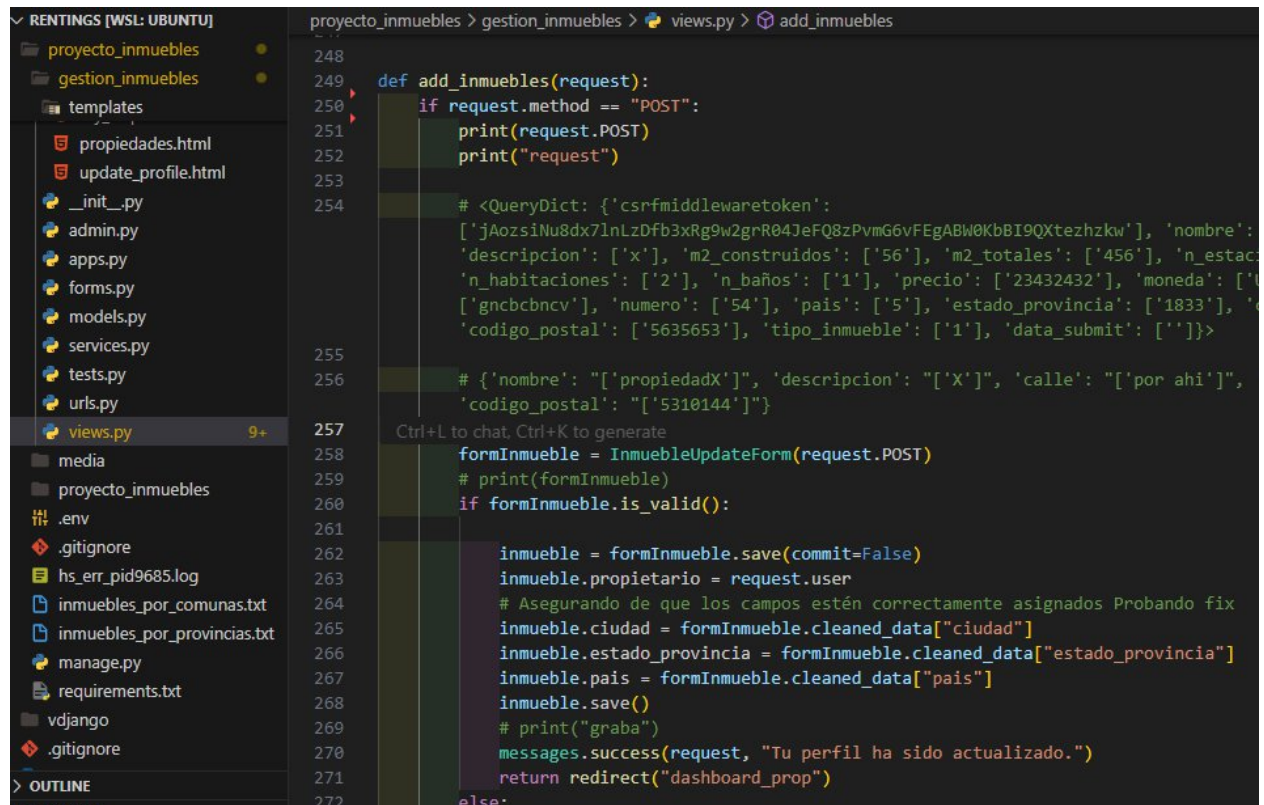


## Hito 4 Parte 2

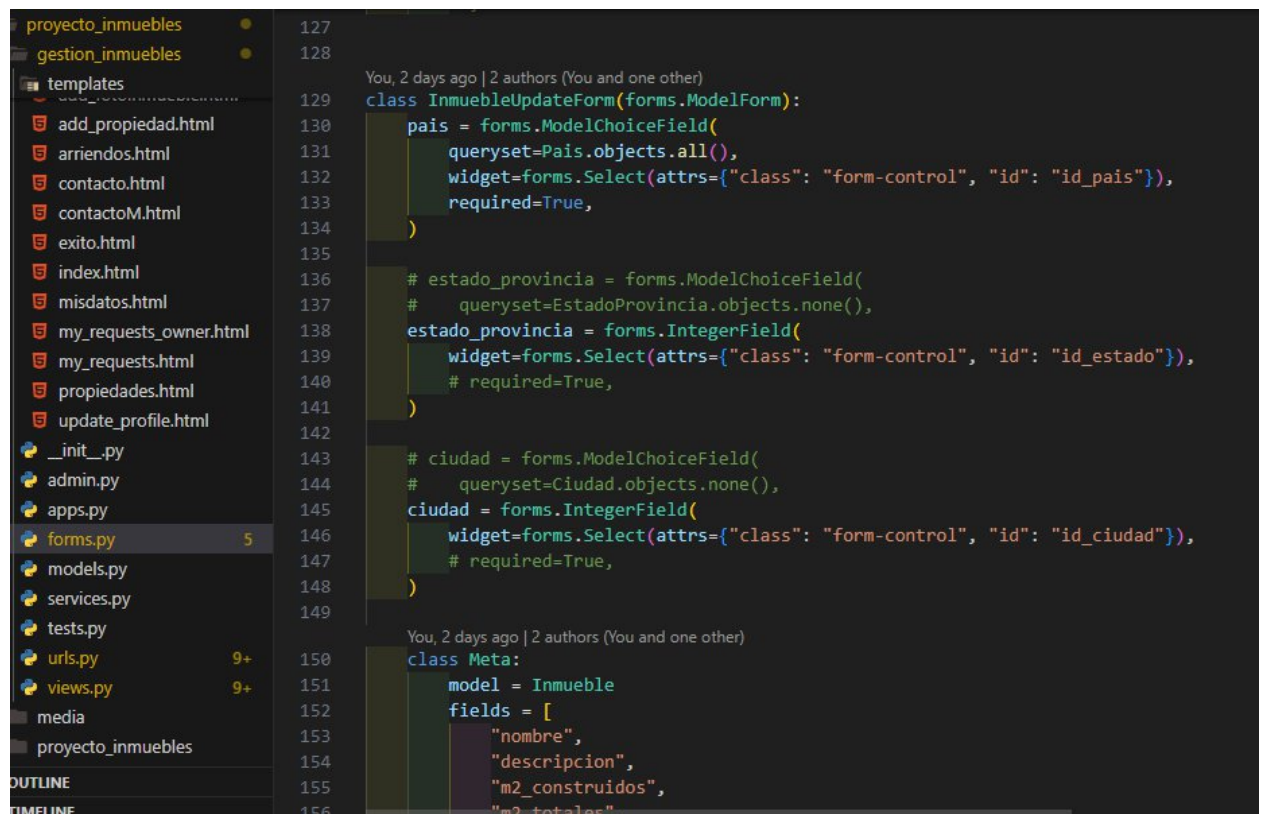
### 1.1 Generar ruta para agregar nueva vivienda



```
projecto_inmuebles > gestion_inmuebles > views.py > add_inmuebles

248
249 def add_inmuebles(request):
250     if request.method == "POST":
251         print(request.POST)
252         print("request")
253
254         # <QueryDict: {'csrfmiddlewaretoken':
255         # ['jAozsiNu8dx71nLzDfb3xRg9w2grR04JefQ8zPvmG6vFEgABW0KbBI9QXtezhkw'], 'nombre':
256         # ['descripcion': ['x'], 'm2_construidos': ['56'], 'm2_totales': ['456'], 'n_estac
257         # ['n_habitaciones': ['2'], 'n_baños': ['1'], 'precio': ['23432432'], 'moneda': ['1
258         # ['gncbcncv'], 'numero': ['54'], 'pais': ['5'], 'estado_provincia': ['1833'], 'c
259         # ['codigo_postal': ['5635653'], 'tipo_inmueble': ['1'], 'data_submit': ['']]>
260
261         # {'nombre': "['propiedadX']", 'descripcion': "['X']", 'calle': "['por ahi']",
262         # 'codigo_postal': "['5310144']"}
263
264         Ctrl+L to chat, Ctrl+K to generate
265         formInmueble = InmuebleUpdateForm(request.POST)
266         # print(formInmueble)
267         if formInmueble.is_valid():
268
269             inmueble = formInmueble.save(commit=False)
270             inmueble.propietario = request.user
271             # Asegurando de que los campos estén correctamente asignados Probandos fix
272             inmueble.ciudad = formInmueble.cleaned_data["ciudad"]
273             inmueble.estado_provincia = formInmueble.cleaned_data["estado_provincia"]
274             inmueble.pais = formInmueble.cleaned_data["pais"]
275             inmueble.save()
276             # print("graba")
277             messages.success(request, "Tu perfil ha sido actualizado.")
278             return redirect("dashboard_prop")
279         else:
```

### 1.2 Genera objeto de formulario:



```
projecto_inmuebles
gestion_inmuebles
templates
add_propiedad.html
arriendos.html
contacto.html
contactoM.html
exito.html
index.html
misdatos.html
my_requests_owner.html
my_requests.html
propiedades.html
update_profile.html
__init__.py
admin.py
apps.py
forms.py
models.py
services.py
tests.py
urls.py
views.py
media
projecto_inmuebles
OUTLINE
TIMELINE

127
128
You, 2 days ago | 2 authors (You and one other)
129 class InmuebleUpdateForm(forms.ModelForm):
130     pais = forms.ModelChoiceField(
131         queryset=Pais.objects.all(),
132         widget=forms.Select(attrs={"class": "form-control", "id": "id_pais"}),
133         required=True,
134     )
135
136     # estado_provincia = forms.ModelChoiceField(
137     #     queryset=EstadoProvincia.objects.none(),
138     estado_provincia = forms.IntegerField(
139         widget=forms.Select(attrs={"class": "form-control", "id": "id_estado"}),
140         # required=True,
141     )
142
143     # ciudad = forms.ModelChoiceField(
144     #     queryset=Ciudad.objects.none(),
145     ciudad = forms.IntegerField(
146         widget=forms.Select(attrs={"class": "form-control", "id": "id_ciudad"}),
147         # required=True,
148     )
149
150     You, 2 days ago | 2 authors (You and one other)
151     class Meta:
152         model = Inmueble
153         fields = [
154             "nombre",
155             "descripcion",
156             "m2_construidos",
157             "m2_totales",
```

```
gestion_inmuebles 150 class Meta:
templates 164     "pais",
add_propiedad.html 165     "estado_provincia",
arriendos.html 166     "ciudad",
contacto.html 167     "codigo_postal",
contactoM.html 168     "tipo_inmueble",
exito.html 169 ]
index.html 170 widgets = {
misdatos.html 171     "nombre": forms.TextInput(attrs={"class": "form-control"}),
my_requests_owner.html 172     "descripcion": forms.TextInput(attrs={"class": "form-control"}),
my_requests.html 173     "m2_construidos": forms.TextInput(attrs={"class": "form-control"}),
propiedades.html 174     "m2_totales": forms.TextInput(attrs={"class": "form-control"}),
update_profile.html 175     "n_estacionamientos": forms.TextInput(attrs={"class": "form-control"}),
_init_.py 176     "n_habitaciones": forms.TextInput(attrs={"class": "form-control"}),
admin.py 177     "n_baños": forms.TextInput(attrs={"class": "form-control"}),
apps.py 178     "precio": forms.TextInput(attrs={"class": "form-control"}),
forms.py 5 179     "moneda": forms.TextInput(attrs={"class": "form-control"}),
models.py 180     "calle": forms.TextInput(attrs={"class": "form-control"}),
services.py 181     "numero": forms.TextInput(attrs={"class": "form-control"}),
tests.py 182     #'ciudad': forms.Select(attrs={'class': 'form-control'}),
urls.py 9+ 183     "codigo_postal": forms.TextInput(attrs={"class": "form-control"}),
views.py 9+ 184     "tipo_inmueble": forms.Select(attrs={"class": "form-control"}),
media 185 }
proyecto_inmuebles 186
LINE 187 def clean(self, *args, **kwargs):
188     print(self.cleaned_data)
189     id_ciudad = self.cleaned_data.get("ciudad")
190     self.cleaned_data.update({"ciudad": Ciudad.objects.get(pk=id_ciudad)})
191     print(self.cleaned_data)
```

### 1.3 Guarda el objeto

```
projecto_inmuebles > gestion_inmuebles > views.py > indice
163 # IT created:
164 #     PerfilUsuario.objects.create(user=instance)
165
166 # @receiver(post_save, sender=User)
167 # def guardar_perfil_usuario(sender, instance, **kwargs):
168 #     instance.perfil.save()
169
170
171 def register(request):
172     if request.method == "POST":
173         form = RegistroForm(request.POST)
174         if form.is_valid():
175             user = form.save()
176             login(request, user)
177             return redirect("indice")
178         else:
179             form = RegistroForm()
180     return render(
181         request, template_name="registration/register.html", context={"form": form}
182     )
183
184
```



```

50 <section id="propiedad">
51 <h2>Propiedad</h2>
52
53 <div class="container">
54     {% if messages %}
55     <div class="alert alert-danger text-center" role="alert">
56         {% for message in messages %}
57             {{ message }}
58         {% endfor %}
59     </div>
60     {% endif %}
61
62     {% if formInmueble.errors %}
63     <div class="alert alert-danger">
64         <ul>
65             {% for field in formInmueble %}
66                 {% for error in field.errors %}
67                     <li>{{ error }}</li>
68                 {% endfor %}
69             {% endfor %}
70         </ul>
71     </div>
72     {% endif %}
73
74     <div class="row justify-content-center">
75         <div class="col-md-6">
76             <div class="card shadow-lg">
77                 <div class="card-header text-center bg-primary text-white">
78                     <h5 class="mb-0">Agregar Data Inmueble</h5>
79                 </div>
80                 <form method="POST">
81                     {% csrf_token %}
82                     {{ formInmueble.as_p }}

```

## 2.1 Generacion de rutas para las vistas de actualizacion de viviendas:

```

urlpatterns = [
    path("", views.indice, name="indice"),
    path("contacto/", views.contacto, name="contacto"),
    path("exito/", views.exito, name="exito"),
    path("arriendos/", views.arriendos, name="arriendos"),
    path("buscar-ciudades/", views.buscar_ciudades, name="buscar_ciudades"),
    path("buscar_inmuebles/", views.buscar_inmuebles, name="buscar_inmuebles"),
    path("registro/", views.register, name="registro"),
    path("misdatos/", views.misdatos, name="misdatos"),
    path("editar/", views.update_profile, name="update_profile"),
    path("dashboard_prop/", views.mostrar_inmuebles, name="dashboard_prop"),
    path("add_propiedad/", views.add_inmuebles, name="agregar_propiedad"),
    path(
        "mostrar_un_inmuebles/<int:inmueble_id>/",
        views.mostrar_un_inmuebles,
        name="mostrar_un_inmuebles",
    ),
    path(
        "request_property/<int:id_propiedad>/",
        views.request_property,
        name="request_property",
    ),
]

```

## 2.2 Generación de objeto de formulario para actualizar/crear/borrar inmueble:

```
120 class FotoInmubleForm(forms.ModelForm):
121     class Meta:
122         model = ImagenInmuble
123         fields = ["imagen"]
124         # widgets = {
125         #     'imagen': forms.ClearableFileInput(attrs={'multiple': False}),
126         # }
127
128
129 class InmubleUpdateForm(forms.ModelForm):
130     pais = forms.ModelChoiceField(
131         queryset=Pais.objects.all(),
132         widget=forms.Select(attrs={"class": "form-control", "id": "id_pais"}),
133         required=True,
134     )
135
136     # estado_provincia = forms.ModelChoiceField(
137     #     queryset=EstadoProvincia.objects.none(),
138     #     widget=forms.Select(attrs={"class": "form-control", "id": "id_estado"}),
139     #     required=True,
140     # )
141
142     # ciudad = forms.ModelChoiceField(
143     #     queryset=Ciudad.objects.none(),
144     #     widget=forms.Select(attrs={"class": "form-control", "id": "id_ciudad"}),
145     #     required=True,
146     # )
147
148     class Meta:
149         model = Inmuble
150         fields = [
151             "nombre",
152             "descripcion",
153             "m2_construidos",
154             "m2_totales",
155             "n_estacionamientos",
156             "n_habitaciones",
157             "n_baños",
158             "precio",
159         ]
160         widgets = {
161             "precio": forms.NumberInput(attrs={"class": "form-control", "id": "id_precio"}),
162         }
```

```
139     widget=forms.Select(attrs={"class": "form-control", "id": "id_estado"}),
140     # required=True,
141
142
143     # ciudad = forms.ModelChoiceField(
144     #     queryset=Ciudad.objects.none(),
145     #     widget=forms.Select(attrs={"class": "form-control", "id": "id_ciudad"}),
146     #     required=True,
147     # )
148
149
150 class Meta:
151     model = Inmuble
152     fields = [
153         "nombre",
154         "descripcion",
155         "m2_construidos",
156         "m2_totales",
157         "n_estacionamientos",
158         "n_habitaciones",
159         "n_baños",
160         "precio",
161     ]
162     widgets = {
163         "precio": forms.NumberInput(attrs={"class": "form-control", "id": "id_precio"}),
164     }
```

```

162         "calle",
163         "numero",
164         "pais",
165         "estado_provincia",
166         "ciudad",
167         "codigo_postal",
168         "tipo_inmueble",
169     ]
170     widgets = {
171         "nombre": forms.TextInput(attrs={"class": "form-control"}),
172         "descripcion": forms.TextInput(attrs={"class": "form-control"}),
173         "m2_construidos": forms.TextInput(attrs={"class": "form-control"}),
174         "m2_totales": forms.TextInput(attrs={"class": "form-control"}),
175         "n_estacionamientos": forms.TextInput(attrs={"class": "form-control"}),
176         "n_habitaciones": forms.TextInput(attrs={"class": "form-control"}),
177         "n_baños": forms.TextInput(attrs={"class": "form-control"}),
178         "precio": forms.TextInput(attrs={"class": "form-control"}),
179         "moneda": forms.TextInput(attrs={"class": "form-control"}),
180         "calle": forms.TextInput(attrs={"class": "form-control"}),
181         "numero": forms.TextInput(attrs={"class": "form-control"}),
182         #'ciudad': forms.Select(attrs={'class': 'form-control'}),
183         "codigo_postal": forms.TextInput(attrs={"class": "form-control"}),
184         "tipo_inmueble": forms.Select(attrs={"class": "form-control"}),
185     }
186
187     def clean(self, *args, **kwargs):
188         print(self.cleaned_data)
189         id_ciudad = self.cleaned_data.get("ciudad")
190         self.cleaned_data.update({"ciudad": Ciudad.objects.get(pk=id_ciudad)})

```

## 2.3 Función para actualizar objeto

```

246 def add_inmuebles(request):
247     if request.method == "POST":
248         print(request.POST)
249         print("request")
250
251         # <QueryDict: {'csrfmiddlewaretoken': ['jAozsiNu8dx7lnLzDfb3xRg9w2grR04JefQ8zPvmG6'],
252         # 'propiedadX': ['x'], 'descripcion': ['x'], 'm2_construidos': ['56'], 'm2_totales': ['454'],
253         # 'n_habitaciones': ['2'], 'n_baños': ['1'], 'precio': ['23432432'], 'moneda': ['USD'],
254         # 'pais': ['5'], 'estado_provincia': ['1833'], 'ciudad': ['565420'], 'codigo_postal': [''],
255         # 'data_submit': ['']}> You, 2 days ago * fix formulario
256
257         # {'nombre': '["propiedadX"]', 'descripcion': '["X"]', 'calle': '["por ahi"]', 'nu
258
259         formInmueble = InmuebleUpdateForm(request.POST)
260         # print(formInmueble)
261         if formInmueble.is_valid():
262
263             inmueble = formInmueble.save(commit=False)
264             inmueble.propietario = request.user
265             # Asegurando de que los campos estén correctamente asignados Probandos fix
266             inmueble.ciudad = formInmueble.cleaned_data["ciudad"]
267             inmueble.estado_provincia = formInmueble.cleaned_data["estado_provincia"]
268             inmueble.pais = formInmueble.cleaned_data["pais"]
269             inmueble.save()
270             # print("graba")
271             messages.success(request, "Tu inmueble ha sido actualizado.")
272             return redirect("dashboard_prop")
273         else:
274             print(formInmueble.errors)
275             messages.error(request, "Hay errores en los datos")
276
277     else:

```





## Robert, Bienvenido a Deux Propiedades

Inicio Dashboard Contacto 23456789-0 Logout

### Mis Inmuebles

Agregar Inmuebles

Departamento

Disponible

Departamento Moderno

Departamento céntrico con terraza  
Precio: 180000.00 USD  
Dirección: Av. Alameda 456, Temuco  
m² construidos: 80  
m² totales: 100  
Estacionamientos: 1  
Habitaciones: 2 | Baños: 2

Modificar

Agregar Foto

Departamento

Disponible

Departamento Moderno

Departamento céntrico con terraza  
Precio: 180000.00 USD  
Dirección: Av. Alameda 456, Temuco  
m² construidos: 80  
m² totales: 100  
Estacionamientos: 1  
Habitaciones: 2 | Baños: 2

Modificar

Agregar Foto

Casa

Disponible

Fun House

very fun house  
Precio: 500000.00 USD  
Dirección: elm street 666, Ciudad de  
Concepcion  
m² construidos: 120  
m² totales: 250  
Estacionamientos: 2  
Habitaciones: 5 | Baños: 2

Modificar

Agregar Foto



Casa

Departamento

Departamento

### 3 Arrendatarios ven ofertas por ciudad

## Luke, Bienvenido a Deux Propiedades

Inicio Arriendos Contacto 15151567-6 Logout

### Búsqueda de Ciudades e Inmuebles

Osorno

Buscar

Osorno - Los Lagos, Chile

### Inmuebles Disponibles

#### Casa de Lujo

Residencia con acabados premium - \$1000000.00 USD

Dirección: Av. Las Palmeras 567, Osorno

Solicitar

Imágenes no disponibles

#### Casa de Lujo

Residencia con acabados premium - \$1000000.00 USD

Dirección: Av. Las Palmeras 567, Osorno

Solicitar

Imágenes no disponibles

### 3.1 Rutas para mostrar inmuebles:

```
propiedades.html
update_profile.html
__init__.py
admin.py
apps.py
forms.py 5
models.py 3
services.py
tests.py
urls.py 9+
views.py 9+
media
proyecto_inmuebles
.env
.gitignore

urlpatterns = [
    path("", views.indice, name="indice"),
    path("contacto/", views.contacto, name="contacto"),
    path("exito/", views.exito, name="exito"),
    path("arriendos/", views.arriendos, name="arriendos"),
    path("buscar-ciudades/", views.buscar_ciudades, name="buscar_ciudades"),
    path("buscar_inmuebles/", views.buscar_inmuebles, name="buscar_inmuebles"),
    path("registro/", views.register, name="registro"),
    path("misdatos/", views.misdatos, name="misdatos"),
    path("editar/", views.update_profile, name="update_profile"),
    path("dashboard_prop/", views.mostrar_inmuebles, name="dashboard_prop"),
    path("add_propiedad/", views.add_inmuebles, name="agregar_propiedad"),
    path(
        "mostrar_un_inmuebles/<int:inmueble_id>/",
        views.mostrar_un_inmuebles,
        name="mostrar_un_inmuebles",
    ),
]
```

### 3.2 Crear vista que permiten ver enlistar inmuebles.

```
gestion_inmuebles
templates
my_requests.html
propiedades.html
update_profile.html
__init__.py
admin.py
apps.py
forms.py 5
models.py 3
services.py
tests.py
urls.py 9+
views.py 9+
media
proyecto_inmuebles
.env
.gitignore
hs_err_pid9685.log
inmuebles_por_comunas.txt
inmuebles_por_provincias.txt
manage.py
requirements.txt
vdjango
OUTLINE
TIMELINE

def mostrar_un_inmuebles(request, inmueble_id):
    inmueble = get_object_or_404(Inmueble, pk=inmueble_id, propietario=request.user)

    if request.method == "POST":
        if "foto_submit" in request.POST:
            form = FotoInmuebleForm(request.POST, request.FILES)
            files = request.FILES["imagen"]  Local variable 'files' is assigned to but never used
            if form.is_valid():
                imagen = form.cleaned_data["imagen"]
                ImagenInmueble.objects.create(propiedad=inmuebles, imagen=imagen)
                messages.success(request, "La foto se ha agregado correctamente.")
                return redirect("dashboard_prop")
            # return redirect(
            #     "mostrar_un_inmuebles", inmueble_id=inmueble_id
            # ) # Redirigir a mostrar detalles
        else:
            messages.error(request, "Hubo un error al intentar guardar la imagen.")

    elif "data_submit" in request.POST:
        form = InmuebleUpdateForm(request.POST, instance=inmuebles)
        if form.is_valid():
            print("entro")
            form.save()
            messages.success(request, "Tu perfil ha sido actualizado.")
            return redirect("mostrar_un_inmuebles", inmueble_id=inmueble_id)
        else:
            messages.error(request, "Hay errores en los datos")
```