* Definimos en el settings.py el LOGIN\_REDURECT\_URL

LOGIN\_REDIRECT\_URL = '/erp/category/list/'

* Creamos un LoginView, el cual como base ya tiene la configuración de login, por lo que solo este código será suficiente para hacer un login

class LoginFormView(LoginView):  
 template\_name = 'login.html'  
  
 def dispatch(self, request, \*args, \*\*kwargs):  
 if request.user.is\_authenticated:  
 return redirect('erp:category\_listview')  
 return super().dispatch(request, \*args, \*\*kwargs)

//Redireccionar después del login al sitio a donde querías ir

<input type="hidden" value="{{ next }}" name="next">

// Login with FormView

class LoginFormView2(FormView):  
 form\_class = AuthenticationForm  
 template\_name = 'login.html'  
 success\_url = reverse\_lazy('erp:category\_listview')  
  
 def dispatch(self, request, \*args, \*\*kwargs):  
 if request.user.is\_authenticated:  
 return redirect('erp:category\_listview')  
 return super().dispatch(request, \*args, \*\*kwargs)  
  
 def form\_valid(self, form):  
 login(self.request, form.get\_user())  
 return HttpResponseRedirect(self.success\_url)

Hacemos logout con LogoutView. Si no agregamos LOGOUT\_REDIRECT\_URL agregamos el next\_page

path('Logout/', LogoutView.as\_view(next\_page='accounts:login'), name='logout'),

--- con una vista

class LogoutRedirectView(RedirectView):  
 pattern\_name = 'accounts:login'  
  
 def dispatch(self, request, \*args, \*\*kwargs):  
 logout(request)  
 return super().dispatch(request, \*args, \*\*kwargs)