

Reporte: Proyecto Final

Heli Saul, Vásquez Gómez, 201700852
*Escuela de Mecánica Eléctrica, Facultad de Ingeniería,
 Universidad de San Carlos de Guatemala*

Resumen: a continuación, se muestra el proyecto el cual consiste en una tienda de cursos, que al seleccionar un curso el estudiante podrá asignarse y pagas dicho curso

B. *Objetivos*

Objetivo General

1. Mostrar el funcionamiento de la pagina

Objetivo especifico

1. Ver los cursos disponibles

```
from typing import Any
from django.shortcuts import render,
redirect
from django.contrib.auth import
authenticate, login
from django.contrib.auth.decorators import
login_required
from django.views.generic import
TemplateView
from django.contrib.auth.models import Group
from django.views import View
from .forms import RegisterForm
from django.views.generic import
TemplateView
# Create your views here.
from django.contrib.auth.views import
PasswordChangeView
from django.contrib.auth.views import
PasswordResetView
from django.urls import reverse_lazy
from django.contrib import messages
from django.contrib.auth import
update_session_auth_hash
from .models import Course

from django.utils.decorators import
method_decorator
```

```
class CustomTemplateView(TemplateView):
    group_name = None

    def get_context_data(self, **kwargs):
        context =
        super().get_context_data(**kwargs)
        user = self.request.user
        if user.is_authenticated:
            group =
            Group.objects.filter(user = user).first()
            if group:
                self.group_name = group.name

        context['group_name']=
        self.group_name
        return context
```

```
class HomeView(CustomTemplateView):

    template_name = 'home.html'

    def get_context_data(self,**kwargs):
        context =
        super().get_context_data(**kwargs)
        user = self.request.user
        group_name = None
        if user.is_authenticated:
            group =
            Group.objects.filter(user=user).first()
            if group:
                group_name = group.name

        context['group_name'] = group_name
        return context
```

```

class PrecingView(CustomeTemplateView):
    template_name = 'precing.html'

class RegisterView(View):
    def get(self, request):
        data = {

            'form': RegisterForm()
        }
        return
render(request, 'registration/register.html',
data)

    def post(self, request):
        user_creation_form =
RegisterForm(data= request.POST)
        if user_creation_form.is_valid():
            user_creation_form.save()
            user = authenticate(username =
user_creation_form.cleaned_data['username'],
password =
user_creation_form.cleaned_data['password1']
)

            login(request, user)
            return redirect('home')

        data = {

            'form': user_creation_form
        }
        return render(request,
'registration/register.html', data)

```

```

class
ProfilePasswordChangeView>PasswordChangeView
):
    template_name =
'registration/change_password.html'
    success_url = reverse_lazy('login')
    def get_context_data(self, **kwargs):
        context =
super().get_context_data(**kwargs)

```

```

        context['password_change'] =
self.request.session.get('password_change',
False)
        return context

    def form_valid(self, form):
        messages.success(self.request, 'Cambi
o de contraseña exitoso')
        update_session_auth_hash(self.reques
t, form.user)
        self.request.session['password_chang
e'] = True
        return super().form_valid(form)

    def form_invalid(self, form):
        messages.error(self.request, 'No se
puede cambiar las contraseña')

        return super().form_invalid(form)

```

```

class ProfileView(CustomeTemplateView):
    template_name = 'profile/profile.html'

    def get_context_data(self, **kwargs):
        context =
super().get_context_data(**kwargs)
        user = self.request.user
        return context

```

mostara los cursos

```

class CoursesView(TemplateView):

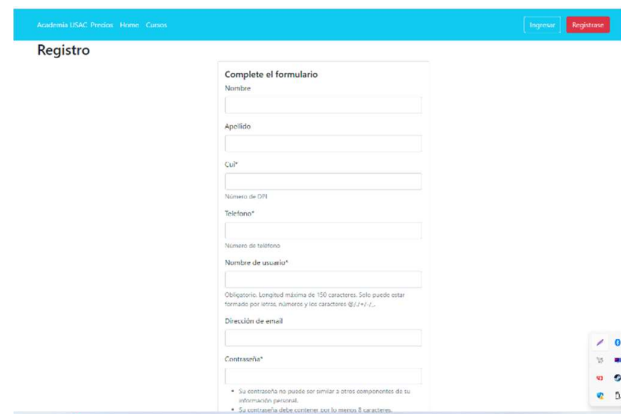
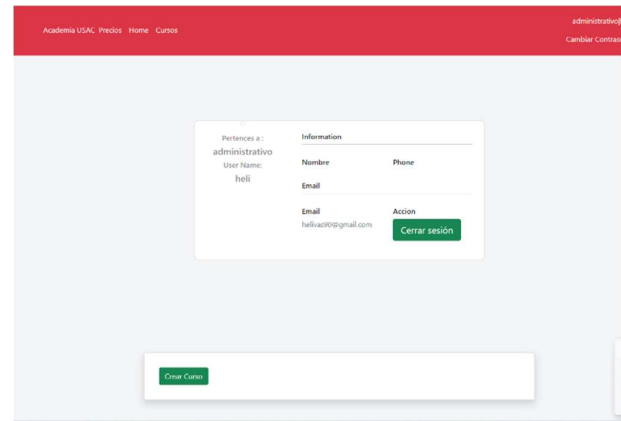
    template_name = 'courses.html'

    def get_context_data(self, **kwargs):
        context
= super().get_context_data(**kwargs)
        courses = Course.objects.all()
        context['courses'] = courses

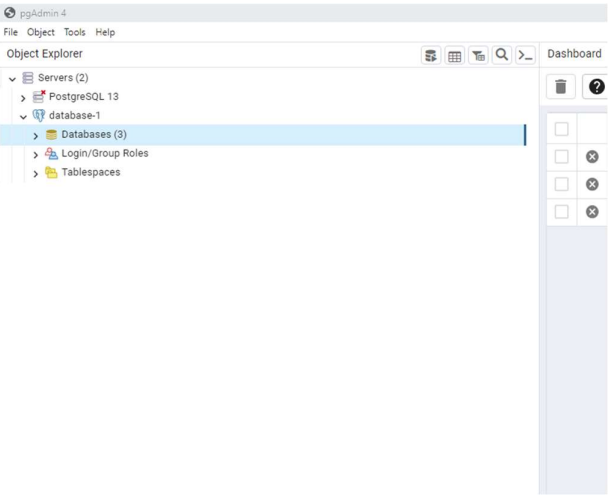
        return context

```

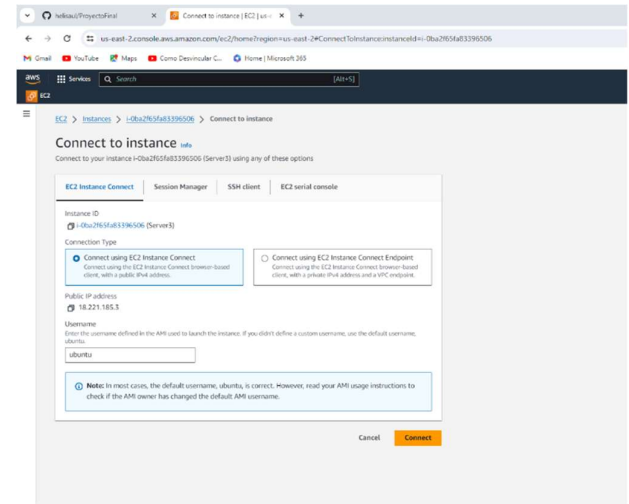
C. Imágenes de la pagina



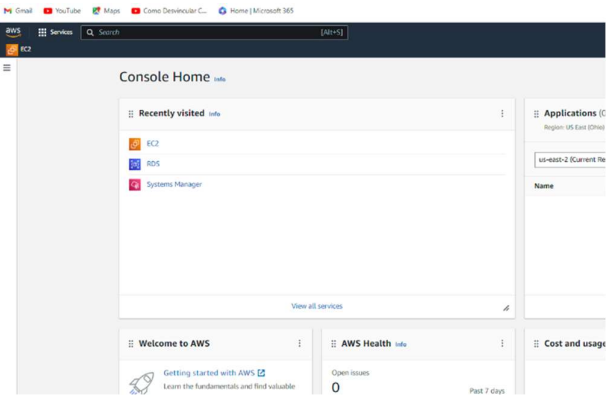
base de datos.



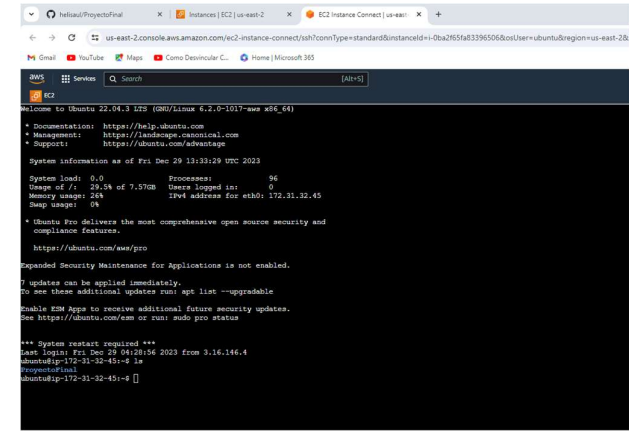
Conexión a mi instancia



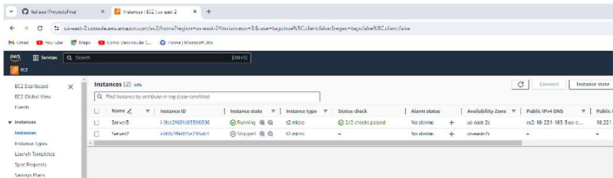
AWS



Mi instancia en ejecucion



Mi Instancia



conclusiones

D.

1. El usuario pueda navegar en la pagina de una manera fácil
2. Se pudo establecer la función en el cual el cliente pueda ver los cursos agregados de parte el administrador
3. Con nuestra instancia en la nube se logro un servidor local en cual funciona similar a una maquina local o pc pero con la única diferencia que cualquiera puede acceder al página.

E.

Código

<https://github.com/helisaul/ProyectoFinal.git>