**Pruebas de Caja Blanca**

**Fisi/documentos/Ajax.py**

def vercomentarios(request):

id\_profe=request.GET['profesor\_id']

1

comentarios=Comentario.objects.filter(Profesor=id\_profe)

data=""

2

for coment in comentarios:

data+="<p><strong><span>Usuario:</strong></span> %s</p> <p><strong><span>Comentario:</strong></span> %s</p> <hr>"%(coment.user,coment.comentario)

3

response={}

4

response['comentarios']=data

return JsonResponse(response)

*Notas: 1 variable*

1

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 1

NO

SI

4

3

2

**Fisi/documentos/Ajax.py**

def get\_curso(request):

id\_area = request.GET['area\_id']

1

cursos = curso.objects.none()

options = '<option value="" selected="selected">---------</option>'

2

if id\_area:

3

cursos = curso.objects.filter(area=id\_area)

4

for curs in cursos:

options += '<option value="%s">%s</option>' % (

5

curs.id,

curs.curso

)

response = {}

response['cursos'] = options

6

print (options)

return JsonResponse(response)

*Notas: 2 variable*

1

NO

SI

4

3

2

-Complejidad ciclomatica: 3

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 2

SI

NO

6

5

**Fisi/documentos/Ajax.py**

def get\_profesor(request):

curso\_id = request.GET['curso\_id']

1

profesores = Profesor.objects.none()

options = '<option value="" selected="selected">---------</option>'

2

5

if curso\_id:

3

profesores = Profesor.objects.filter(curso=curso\_id)

4

for profe in profesores:

options += '<option value="%s">%s</option>' % (

profe.id,

profe.nombre

)

response = {}

response['profesores'] = options

6

return JsonResponse(response)

*Notas: 2 variable*

1

NO

SI

4

3

2

-Complejidad ciclomatica: 3

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 2

SI

NO

6

5

**Fisi/documentos/Ajax.py**

def get\_datos(request):

1

id\_profesor= request.GET['profesor\_id']

profesores=Profesor.objects.filter(id=id\_profesor)

data=""

2

for pro in profesores:

data+='<p>Nombre : %s </p> <p>Apellido : %s </p><p>Email : %s'%(pro.nombre,pro.apellido,pro.email)

3

response={}

4

response['profesores']=data

return JsonResponse(response)

*Notas: 1 variable*

1

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 1

NO

SI

4

3

2

**Fisi/documentos/Ajax.py**

2

1

def archivoxarea(request):

id\_area=request.GET["id"]

archivos=Archivo.objects.filter(area=id\_area).order\_by("-timestamp")

data=""

for archi in archivos:

data+="<div class='panel panel-primary row'><div class='panel-heading'><p><strong>Autor:%s</strong></p></div><div class='panel-body'><div class='sub\_left'><p><strong> Nombre Archivo:</strong><span> %s</span></p><p><strong> Area:</strong><span>%s</span></p><p><strong>Curso: </strong><span>%s</span></p><p> <strong>Profesor: </strong><span>%s</span></p><hr> <a href='/static/%s' class='btn ' role='button' target='\_blank'> Ver </a></div></div></div></div>"%(archi.user.username,archi.nombre,archi.area,archi.curso,archi.profesor,archi.archivo)

3

response={}

4

response['archivos']=data

return JsonResponse(response)

*Notas: 1 variable*

1

NO

SI

4

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 1

**Fisi/documentos/Ajax.py**

2

1

def archivoxcurso(request):

id\_curso=request.GET["id"]

archivos=Archivo.objects.filter(curso=id\_curso).order\_by("-timestamp")

data=""

for archi in archivos:

data+="<div class='panel panel-primary row'><div class='panel-heading'><p><strong>Autor:%s</strong></p></div><div class='panel-body'><div class='sub\_left'><p><strong> Nombre Archivo:</strong><span> %s</span></p><p><strong> Area:</strong><span>%s</span></p><p><strong>Curso: </strong><span>%s</span></p><p> <strong>Profesor: </strong><span>%s</span></p><hr> <a href='/static/%s' class='btn ' role='button' target='\_blank'> Ver </a></div></div></div></div>"%(archi.user.username,archi.nombre,archi.area,archi.curso,archi.profesor,archi.archivo)

3

4

response={}

response['archivos']=data

return JsonResponse(response)

*Notas: 1 variable*

1

NO

SI

4

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 1

**Fisi/documentos/Ajax.py**

2

1

def archivoxprofesor(request):

id\_profesor=request.GET["id"]

archivos=Archivo.objects.filter(profesor=id\_profesor).order\_by("-timestamp")

data=""

for archi in archivos:

data+="<div class='panel panel-primary row'><div class='panel-heading'><p><strong>Autor:%s</strong></p></div><div class='panel-body'><div class='sub\_left'><p><strong> Nombre Archivo:</strong><span> %s</span></p><p><strong> Area:</strong><span>%s</span></p><p><strong>Curso: </strong><span>%s</span></p><p> <strong>Profesor: </strong><span>%s</span></p><hr> <a href='/static/%s' class='btn ' role='button' target='\_blank'> Ver </a></div></div></div></div>"%(archi.user.username,archi.nombre,archi.area,archi.curso,archi.profesor,archi.archivo)

3

response={}

4

response['archivos']=data

return JsonResponse(response)

*Notas: 1 variable*

1

NO

SI

4

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 1

**Fisi/documentos/Validators.py**

1

*def valid\_extension(value):*

2

*if (not value.name.endswith('.pdf')):*

3

*raise ValidationError("Solo archivos PDF")*

*Notas: 1 variable*

1

NO

SI

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 2

4

**Fisi/documentos/views.py**

@login\_required

def subir\_archivos(request):

form = DocumentoA(request.POST,request.FILES)

1

titulo=""

context={"form":form,

"titulo":titulo}

2

if form.is\_valid():

instance = form.save(commit=False)

nombre=form.cleaned\_data.get("nombre")

instance.user = request.user

instance.save()

3

form.save()

form = DocumentoA()

context={

"titulo":"El archivo se subio correctamente",

"form":form

}

4

return render(request,'Documentos/subir\_archivos.html',context)

*Notas: 1 variable*

1

NO

SI

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 2

4

**Fisi/documentos/views.py**

def mostrarprofesores(request):

1

profesores=Profesor.objects.all()

form= ComentarioC(request.POST)

2

if form.is\_valid():

comentario=form.save(commit=False)

comentario.user=request.user

3

comentario.save()

form.save()

4

return render(request,'Profesores/mostrarprofesor.html',{'profesores':profesores})

*Notas: 1 variable*

1

NO

SI

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 2

4

**Fisi/inicio/form.py/class EditarEmailForm**

def clean\_email(self):

email = self.cleaned\_data['email']

1

# Comprobar si ha cambiado el email

actual\_email = self.request.user.email

username = self.request.user.username

2

if email != actual\_email:

# Si lo ha cambiado, comprobar que no exista en la db.

3

# Exluye el usuario actual.

existe = User.objects.filter(email=email).exclude(username=username)

4

if existe:

5

raise forms.ValidationError('Ya existe un email igual en la db.')

6

return email

*Notas: 3 variables*

1

NO

SI

3

2

-Complejidad ciclomatica:3

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 3

NO

4

SI

5

6

**Fisi/inicio/form.py/class EditarContresenaForm**

def clean\_password2(self):

1

password = self.cleaned\_data['password']

password2 = self.cleaned\_data['password2']

2

if password != password2:

3

raise forms.ValidationError('Las contraseñas no coinciden.')

4

return password2

*Notas: 2 variable*

1

NO

SI

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 1
* Cobertura de decisión: 2

4

**Fisi/inicio/views.py**

@login\_required

def editar(request):

form = EditarContrasenaForm(request.POST)

1

context={

"form":form,

}

2

if form.is\_valid():

request.user.password = make\_password(form.cleaned\_data['password'])

request.user.save()

3

context={

"titulo":"La contraseña se cambio con exito",

}

return render(request, 'editar.html', context)

else:

4

form = EditarContrasenaForm()

5

return render(request, 'editar.html', context)

*Notas: 1 variable*

1

NO

SI

3

2

-Complejidad ciclomatica: 2

- Pruebas minimas para lograr

* Cobertura de sentencia: 2
* Cobertura de decisión: 2

4

5