

Informe de Laboratorio 06

Tema: Django (Admin)

Nota

Integrantes	Escuela	Asignatura
Miguel Angel Alvarez Choque 20230477 Rodrigo Alexander Fernández Huarca 20230465 Eduardo Joel Cuno Salazar 20231497 Jose Maria Ticona Saure 20233482	Escuela Profesional de Ingeniería de Sistemas	Programación Web II Semestre: I

Laboratorio	Tema	Duración
06	Django (Admin)	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 05 de junio 2024	Al 07 de junio 2024

RESULTADOS Y PRUEBAS

1. EJERCICIOS RESUELTOS:

- Como parte del trabajo se desarrollo todas las actividades de la siguiente página web. Se dara detalle a cada parte de la introducción a *Django*.
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website

1.1. Parte 1:

- En esta parte se explica lo que aprenderá y proporciona una descripción general del sitio web de ejemplo de "biblioteca local.^{en} el que trabajaremos y evolucionaremos en artículos posteriores.

1.2. Parte 2:

- En este segundo artículo se muestra cómo podemos crear un proyecto de sitio web .esqueleto como base, que luego puedes completar con configuraciones, rutas, modelos, vistas y plantillas específicas del sitio.

1.2.1. Pasos realizados:

- creación del proyecto:

```
(entorno-DJANGO) C:\Users\Eduardo\Desktop\entorno-virtual\locallibrary>django-admin startproject locallibrary
```

- creación de la aplicación catálogo:

```
(entorno-DJANGO) C:\Users\Eduardo\Desktop\entorno-virtual\locallibrary\locallibrary>py manage.py startapp catalog
(entorno-DJANGO) C:\Users\Eduardo\Desktop\entorno-virtual\locallibrary\locallibrary>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

- Añadiendo la aplicación a urls.py:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    #añadiendo CatalogConfig
    'catalog.apps.CatalogConfig',
]
```

- Primera prueba. Se da un error previsto:

Page not found (404)

Request Method: GET
Request URL: http://127.0.0.1:8000/catalog/

Using the URLconf defined in locallibrary.urls, Django tried these URL patterns, in this order:

1. admin/
- 2.
3. ^static/(?P<path>.*)\$

The current path, catalog/, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

1.3. Parte 3:

- En esta parte se muestra cómo definir modelos para el sitio web *LocalLibrary*. Se explica qué es un modelo, cómo se declara y algunos de los principales tipos de campos. También muestra brevemente algunas de las principales formas en que puede acceder a los datos del modelo..

1.3.1. Pasos realizados:

- Migración de los modelos a la base de datos:

```
(entorno-DJANGO) C:\Users\Eduardo\Desktop\entorno-virtual\locallibrary\locallibrary>python manage.py makemigrations
Migrations for 'catalog':
  catalog\migrations\0002_genre_alter_author_options_author_date_of_death_book_and_more.py
    - Create model Genre
    - Change Meta options on author
    - Add field date_of_death to author
    - Create model Book
    - Create model BookInstance
    - Create constraint genre_name_case_insensitive_unique on model genre
    - Add field genre to book

(entorno-DJANGO) C:\Users\Eduardo\Desktop\entorno-virtual\locallibrary\locallibrary>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, catalog, contenttypes, sessions
Running migrations:
  Applying catalog.0002_genre_alter_author_options_author_date_of_death_book_and_more... OK
```

1.4. Parte 4:

- En esta parte usaremos el sitio *Django Admin* para agregar algunos datos de libros reales”. Primero, se registro los modelos en el sitio de administración, luego se inició sesión y creó algunos datos.

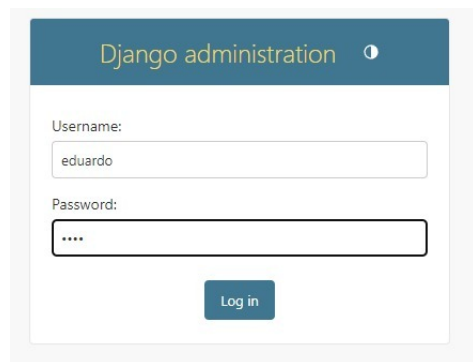
1.4.1. Pasos realizados:

- creando un superusuario:

```
(entorno-DJANGO) C:\Users\Eduardo\Desktop\entorno-virtual\locallibrary\locallibrary>python manage.py createsuperuser
Username (leave blank to use 'eduardo'): eduardo
Email address: ecunos@unsa.edu.pe
Password:
Password (again):
```

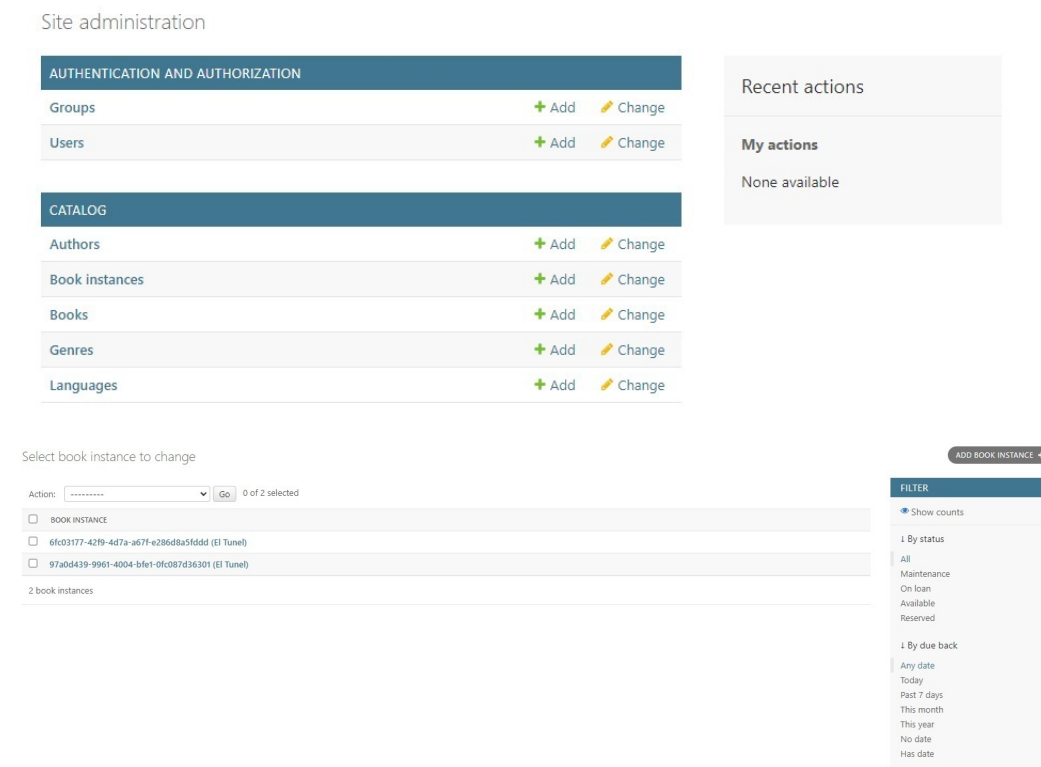
- Para este caso se creó el superusuario con estos datos:

- username: eduardo
- Password: 1234



The screenshot shows the Django administration login interface. It has a blue header with 'Django administration' and a user icon. Below the header, there are two input fields: 'Username:' with 'eduardo' entered, and 'Password:' with four dots. A 'Log in' button is at the bottom.

- añadiendo filtros a las listas:



The screenshot shows the 'Site administration' page in Django. It has a blue header with 'Django administration'. Below the header, there are two main sections: 'AUTHENTICATION AND AUTHORIZATION' and 'CATALOG'. The 'AUTHENTICATION AND AUTHORIZATION' section has links for 'Groups' and 'Users', each with '+ Add' and 'Change' buttons. The 'CATALOG' section has links for 'Authors', 'Book instances', 'Books', 'Genres', and 'Languages', each with '+ Add' and 'Change' buttons. On the right side, there is a 'Recent actions' section with 'My actions' and 'None available'. At the bottom, there is a 'Select book instance to change' section with a table of book instances. The table has columns for 'Action', 'Book instance', and 'Status'. There are two book instances listed: one with ID '6fc03177-42f9-4d7a-a67f-e286d8a5fddd' and another with ID '97a0d439-9961-4004-b1e1-0fc087d36301'. Both are marked as '2 book instances'. On the far right, there is a 'FILTER' section with 'Show counts' and 'By status' filters. The 'By status' filter has options: 'All', 'Maintenance', 'On loan', 'Available', and 'Reserved'. The 'By due back' filter has options: 'Any date', 'Today', 'Past 7 days', 'This month', 'This year', 'No date', and 'Has date'.

1.5. Parte 5:

- En esta parte se agregó el código que muestra nuestra primera página completa: una página de inicio para el sitio web *LocalLibrary*. La página de inicio mostrará la cantidad de registros que tenemos para cada tipo de modelo y proporcionará enlaces de navegación en la barra lateral a nuestras otras páginas.

1.5.1. Pasos realizados:

- creando la página web:

[Home](#)
[All books](#)
[All authors](#)

Local Library Home

Welcome to LocalLibrary, a website developed by Mozilla Developer Network!

Dynamic content

The library has the following record counts:

- **Books:** 1
- **Copies:** 2
- **Copies available:** 0
- **Authors:** 1

Información adicional

Número total de géneros: 1

Número de libros que contienen la palabra "python": 0

1.6. Parte 6:

- En esta parte se agregó listas y páginas de detalles para libros y autores. Además, vimos sobre las vistas genéricas basadas en clases y analizamos cómo se puede reducir la cantidad de código que debe escribir para casos de uso comunes.

1.7. Parte 7:

- En esta parte se agrega un contador de visitas basado en sesiones a la página de inicio.

1.7.1. Pasos realizados:

- Se crea el contador de veces que se visitó la página:

[Inicio](#)

[Todos los libros](#)

[Todos los](#)

[autores](#)

Local Library Home

Welcome to LocalLibrary, a website developed by *Mozilla Developer Network!*

Dynamic content

The library has the following record counts:

- **Books:** 0
- **Copies:** 0
- **Copies available:** 0
- **Authors:** 0

You have visited this page 6 times.

Información adicional

Número total de géneros:

1.8. Parte 8:

- En esta parte se preparó para que los usuarios inicien sesión en su sitio con sus propias cuentas y cómo controlar lo que pueden hacer y ver en función de si han iniciado sesión o no y de sus permisos .

1.8.1. Pasos realizados:

- Se crea el inicio de sesión:

[Inicio](#)
[Todos los libros](#)
[Todos los autores](#)

Username:
Password:

[¿Olvidó su contraseña?](#)

- Se crea el sistema para restablecer contraseña:

[Inicio](#)
[Todos los libros](#)
[Todos los autores](#)

Restablecer contraseña

- Se crea el envío de correos para restablecer contraseña:

[Inicio](#)
[Todos los libros](#)
[Todos los autores](#)

Le hemos enviado por correo electrónico instrucciones para configurar su contraseña. Si no han llegado en unos minutos, revisa tu carpeta de spam.

1.9. Parte 9:

- En esta parte se propuso trabajar con formularios HTML en Django y, en particular, la forma más sencilla de poder escribir formularios para crear, actualizar y eliminar instancias de modelo.

1.9.1. Pasos realizados:

- Se crea el envío de correos para restablecer contraseña:

```
locallibrary-ejrcio1 > catalog > book_renew_librarian.html > form > input
1  {% extends "base_generic.html" %}
2
3  {% block content %}
4  <h1>Renew: {{ book_instance.book.title }}</h1>
5  <p>Borrower: {{ book_instance.borrower }}</p>
6  <p>{% if book_instance.is_overdue %} class="text-danger">Due date: {{ book_instance.due_date }}</p>
7
8  <form action="" method="post">
9  <input type="hidden" value="{{ csrf_token }}" name="csrf_token">
10 <table>
11 <tr>
12 <td>{{ form.as_table }}
13 </td>
14 </tr>
15 </table>
16 </form>
17
```

1.10. Parte 10:

- En esta parte se requerirán más cambios para garantizar que todo siga funcionando y no se introduzcan errores a medida que se realizan más cambios. En esta parte vamos a buscar cómo automatizar las pruebas unitarias de su sitio web utilizando el marco de prueba de Django.

1.10.1. Pasos realizados:

- Se crea el envío de correos para restablecer contraseña:

```
locallibrary-ejrcio1 > catalog > urls.py > ...
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5  path("", views.index, name="index"),
6  path("books/", views.BookListView.as_view(), name="books"),
7
8  ]
9
10 urlpatterns += [
11 path("book/<uid:pk>/renew/", views.renew_book_librarian, name="renew-book-librarian"),
12 ]
13
14 urlpatterns += [
15 path("author/create/", views.AuthorCreate.as_view(), name="author-create"),
16 path("author/<int:pk>/update/", views.AuthorUpdate.as_view(), name="author-update"),
17 path("author/<int:pk>/delete/", views.AuthorDelete.as_view(), name="author-delete"),
18 ]
19
```

1.11. Parte 11:

- En esta parte se buscará encontrar un host para implementar nuestro sitio web y hacer todo lo posible para que el sitio esté listo para la producción.

1.11.1. Pasos realizados:

- Se crea el envío de correos para restablecer contraseña:

```
locallibrary-ejrcicio1 > catalog > templates > catalog > author_confirm_delete.html > form
1 {% extends "base_generic.html" %}
2 {% block content %}
3 <h1>Delete Author: {{ author }}</h1>
4 {% if author.book_set.all %}
5 <p>You can't delete this author until all their books have been deleted:</p>
6 <ul>
7   {% for book in author.book_set.all %}
8     <li><a href="{% url 'book-detail' book.pk %}">{{book}}</a> ({{book.bookinstance_set.all.count}})</li>
9   {% endfor %}
10 </ul>
11 {% else %}
12 <p>Are you sure you want to delete the author?</p>
13 <form action="" method="POST">
14   {% csrf_token %}
15   <input type="submit" action="" value="Yes, delete.">
16 </form>
17 {% endif %}
18 {% endblock %}
```

1.12. Partes importantes del código:

- A pesar de que la parte final del proyecto no resultara se llevo a implementar el código de manera optima. A continuación se comparte las partes más importantes del código:

```
locallibrary-ejrcicio1 > catalog > book_renew_librarian.html > form > input
1 {% extends "base_generic.html" %}
2
3 {% block content %}
4 <h1>Renew: {{ book_instance.book.title }}</h1>
5 <p>Borrower: {{ book_instance.borrower }}</p>
6 <p>{% if book_instance.is_overdue %} class="text-danger">Due date: {{ book_instance.due_back }}</p>
7
8 <form action="" method="post">
9   {% csrf_token %}
10   <table>
11     {{ form.as_table }}
12   </table>
13   <input type="submit" value="Submit">
14 </form>
15 {% endblock %}
```

```
locallibrary-ejrcicio1 > catalog > urls.py ...
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index, name='index'),
6     path('books/', views.BookListView.as_view(), name='books'),
7
8 ]
9 urlpatterns += [
10     path('book/<uuid:pk>/renew/', views.renew_book_librarian, name='renew-book-librarian'),
11 ]
12 urlpatterns += [
13     path('author/create/', views.AuthorCreate.as_view(), name='author-create'),
14     path('author/<int:pk>/update/', views.AuthorUpdate.as_view(), name='author-update'),
15     path('author/<int:pk>/delete/', views.AuthorDelete.as_view(), name='author-delete'),
16 ]
17
```

```
locallibrary-ejrcicio1 > catalog > templates > catalog > autor_detail.html > ...
1 {% block sidebar %}
2 {{ block.super }}
3
4 {% if perms.catalog.change_author or perms.catalog.delete_author %}
5 <hr>
6 <ul class="sidebar-nav">
7   {% if perms.catalog.change_author %}
8     <li><a href="{% url 'author-update' author.id %}">Update author</a></li>
9   {% endif %}
10   {% if not author.book_set.all and perms.catalog.delete_author %}
11     <li><a href="{% url 'author-delete' author.id %}">Delete author</a></li>
12   {% endif %}
13 </ul>
14 {% endif %}
15
16 {% endblock %}
17
```

```
locallibrary-ejrcicio1 > catalog > tests > test_forms.py > RenewBookFormTest
1 import datetime
2 from django.test import TestCase
3 from django.utils import timezone
4 from catalog.forms import RenewBookForm
5 class RenewBookFormTest(TestCase):
6     def test_renew_form_date_field_label(self):
7         form = RenewBookForm()
8         self.assertTrue(form.fields['renewal_date'].label is None or form.fields['renewal_date'].label == 'renewal date')
9     def test_renew_form_date_field_help_text(self):
10         form = RenewBookForm()
11         self.assertEqual(form.fields['renewal_date'].help_text, 'Enter a date between now and 4 weeks (default is 4 weeks)')
12     def test_renew_form_date_in_past(self):
13         date = datetime.date.today() - datetime.timedelta(days=1)
14         form = RenewBookForm(data={'renewal_date': date})
15         self.assertFalse(form.is_valid())
16     def test_renew_form_date_too_far_in_future(self):
17         date = datetime.date.today() + datetime.timedelta(weeks=4) + datetime.timedelta(days=1)
18         form = RenewBookForm(data={'renewal_date': date})
19         self.assertFalse(form.is_valid())
20     def test_renew_form_date_today(self):
21         date = datetime.date.today()
22         form = RenewBookForm(data={'renewal_date': date})
```

CUESTIONARIO:

Por cada integrante del equipo, resalte un aprendizaje que adquirió al momento de estudiar esta primera parte de Django (Admin). No se reprima de ser detallista. Coloque su nombre entre parentesis para saber que es su aporte.

- **Eduardo Joel Cuno Salazar:** Un aprendizaje destacado al estudiar la primera parte de Django es la capacidad de configurar y personalizar la interfaz de administración para gestionar eficientemente los datos de la aplicación. Es muy intuitivo y completo, lo que permite ahorrar mucho tiempo y mejorar significativamente la experiencia del usuario. Este conocimiento nos permite crear aplicaciones web robustas y fáciles de entender, mejorando así la eficiencia del desarrollo.
- **Rodrigo Alexander Fernández Huarca :** Aprender Django es una experiencia rica y gratificante, ya que te permite desarrollar aplicaciones web robustas y escalables rápidamente. Su sistema de migraciones, la modularidad y extensibilidad sin mencionar la seguridad que brinda a nuestros proyectos son pilares para que podamos crear webs bien adaptadas a los requerimientos actuales. También resaltan las plantillas, la documentación existente y la enorme comunidad que posee.
- **Jose Maria Ticona Saure:** Django hace que hacer formularios y manejar datos en tu sitio web sea más fácil. Con Django, no tienes que preocuparte tanto por cómo hacer que tus formularios se vean bien o cómo verificar si la información que ingresan los usuarios es correcta. También te da formas rápidas de crear páginas donde puedas agregar, editar o borrar cosas como publicaciones o usuarios.
- **Miguel Angel Alvarez Choque:** Aprender Django me impresionó en la forma en que se divide las partes de la programación web. La división en los componentes en archivos `.py` como la configuración, modelos, así como el archivo SQLite facilita la organización de un proyecto web. Tengo claro que Django nos permite construir aplicaciones web de manera rápida y efectiva, aprovechando de esta forma muchas funcionalidades integradas y una arquitectura bien pensada. Me queda claro que lo usare ahora que empiezo en la carrera así como también en el momento que ya sea un profesional.

CONCLUSIONES

- Aprender Django te equipa con las habilidades necesarias para desarrollar aplicaciones web modernas y eficientes. Su enfoque en la simplicidad, la rapidez de desarrollo y la seguridad lo convierte en una excelente elección tanto para principiantes como para desarrolladores experimentados. Además, la comunidad y la documentación de Django proporcionan un soporte invaluable durante el proceso de aprendizaje y desarrollo. Con Django, puedes construir aplicaciones web de manera rápida, segura y escalable, lo que te permite centrarte en la creación de valor a través de tu código

METODOLOGÍA DE TRABAJO

- Para el presente trabajo se repartió cada parte del trabajo a un integrante del equipo. Los roles principales fueron:
 - Desarrollo de las actividades y ejercicios del laboratorio: Eduardo, Rodrigo y José Maria.
 - Elaboración del informe: Miguel

REFERENCIAS Y BIBLIOGRAFÍA

Para el trabajo se ocuparon fuentes básicas para entender el uso de Django. Se procede a compartir las fuentes bibliograficas:

- <https://github.com/mdn/django-locallibrary-tutorial>
- <https://github.com/rescobedoq/pw2/tree/main/labs/lab05>
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website

1.13. URL'S del repositorio:

- URL del Repositorio GitHub donde se elaboró el trabajo del laboratorio.
- <https://github.com/ELGRANn/Pweb-6.git>
- URL personal de cada integrante del grupo.
 - Miguel Angel Alvarez Choque:
https://github.com/miguelnodjan/pw2_24a.git
 - Eduardo Joel Cuno Salazar:
<https://github.com/ELGRANn/pw2-24a.git>
 - Rodrigo Alexander Fernández Huarca:
<https://github.com/RdrigoFH/pw2-24a.git>
 - Jose Maria Ticona Saure:
<https://github.com/joseticonasaure/pw2-24a.git>

1.14. Estructura de laboratorio 5

```
D:.\
|-----evidencias
+-----locallibrary-ejercicio1
|-----catalog
|       |-----migrations
|       +-----__pycache__
|       |-----static
|       +-----css
|       |-----templates
|       +-----catalog
|       +-----__pycache__
|-----locallibrary
|       +-----__pycache__
|-----templates
|-----registration
```

2. Rúbricas

2.1. Entregable Informe

Tabla 2: Tipo de Informe

Informe		
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.	Nota
Observaciones	Respetar la estructura de organización para la ubicación de los entregables. Por cada observación dentro del informe se le descontará puntos. Se debe incluir el código fuente latex del informe	

2.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 3: Niveles de desempeño

Nivel				
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 4: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	1	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	x	2	
Total		20		12	