

Universidad Tecnológica Nacional



Trabajo Práctico Integrador Soporte a la Gestión de Datos con Programación Visual TPI-Cursado 2023

Comisión N° 401

Grupo 9

Integrantes:

Glocer, Martin	48165
Hogan, Roy	42694
Nasatsky, Mijael	48100
Velazquez, Francisco	48280

Profesores: Mario Castagnino - Juan Ignacio Torres

Índice

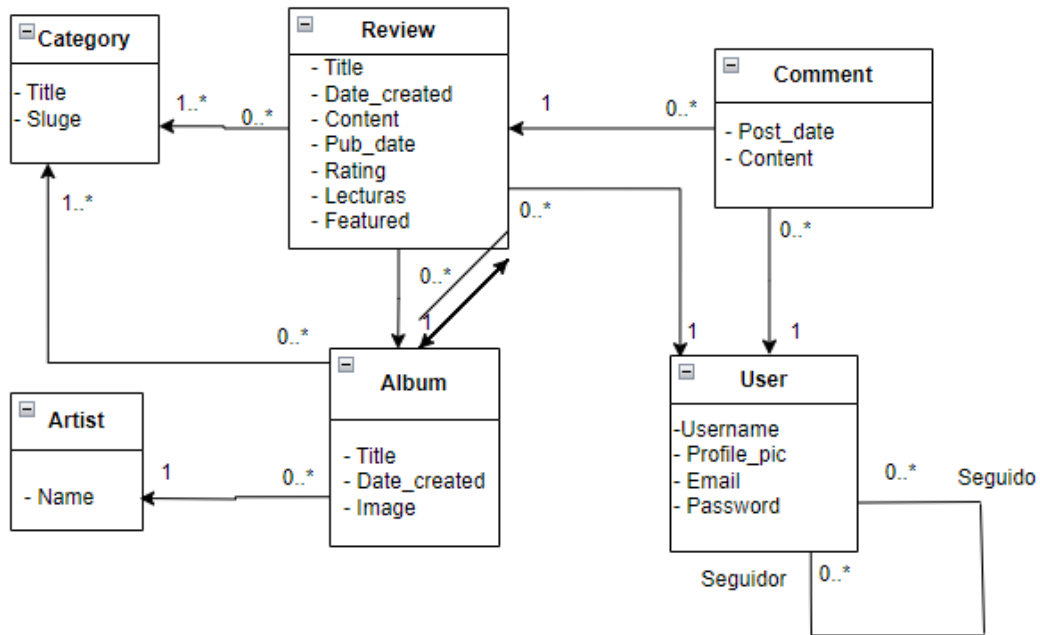
Nombre del Proyecto: MusiBlog.....	2
Narrativa.....	2
Modelo de Dominio.....	3
Reglas de negocio del sistema.....	4
Casos de Uso.....	5
Requerimientos Funcionales.....	11
Requerimientos No Funcionales.....	11
Cálculos.....	12
Stack Tecnológico.....	15
Link Código fuente.....	17

Nombre del Proyecto: MusiBlog

Narrativa

MusiBlog es un sistema diseñado para brindar a los amantes de la música una experiencia única y enriquecedora. Esta plataforma representa una fusión perfecta entre una red social y una plataforma de reseñas de álbumes musicales. El proyecto se desarrolla con el objetivo de conectar a los usuarios a través de su pasión por la música, ofreciéndoles una plataforma dinámica y atractiva para compartir opiniones, descubrir nueva música y tomar decisiones informadas sobre qué álbumes escuchar a continuación.

Modelo de Dominio



Reglas de negocio del sistema

RN001: El usuario puede seguir a una cantidad ilimitada de usuarios, pero no a él mismo.

RN002: El usuario debe estar registrado para poder realizar reseñas y/o comentarios.

RN003: Un usuario se identifica con un ID único y además se registra contraseña, nombre, mail y foto de perfil

RN004: Las reseñas de los usuarios deben incluir un título y un cuerpo de texto con un mínimo de caracteres para garantizar que las reseñas sean informativas.

RN005: Las calificaciones de los álbumes deben estar en un rango específico, por ejemplo, de 1 a 5 estrellas, para mantener la uniformidad en las calificaciones.

RN006: Los usuarios pueden editar y eliminar sus propias reseñas, pero no pueden modificar las reseñas de otros usuarios.

RN007: No pueden existir dos usuarios que tengan el mismo Mail.

RN008: Un Usuario puede no hacer Reseñas, pero una Reseña si o si le pertenece a un Usuario.

RN009: Un Álbum puede tener más de una categoría.

RN010: Las contraseñas de los usuarios deben ser de al menos 8 caracteres y no enteramente numéricas.

Casos de Uso

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Dialogal

Nombre: CUUS01_RealizarReseña

Meta: Realizar una reseña

Actores: Usuario

Precondición: El usuario está registrado y logueado.

Disparador: Usuario clickea la opción realizar reseña

Camino básico:

1. El usuario selecciona un álbum
2. Sistema muestra los datos de dicho álbum
3. Usuario selecciona la opción realizar reseña
4. Sistema abre cuadro de texto para ingresar la reseña
5. Usuario confirma la reseña
6. Sistema muestra cartel informando que la reseña se creó exitosamente

Camino alternativo:

5.a<previo> Usuario cancela la reseña

5.a.1 Fin de CU.

Postcondición:

Éxito: La reseña quedó registrada

Éxito alternativo: <vacío>

Fracaso: El usuario no hizo la reseña

Reglas de negocio:

RN002:El usuario debe estar registrado para poder realizar reseñas y/o comentarios.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Dialogal

Nombre: CUUS02_RegistrarUsuario

Meta: Registrar nuevo usuario

Actores: Usuario

Precondición: -

Disparador: Usuario clickea la opción Crear una cuenta

Camino básico:

1. El usuario clickea Crear una cuenta
2. El sistema solicita un nombre de usuario, un mail, una contraseña y una imagen de perfil.
3. El Usuario ingresa los datos solicitados.
4. El Sistema valida el mail y la contraseña ingresada.
5. Usuario confirma el registro.
6. Sistema muestra cartel informando que el registro se realizó exitosamente.

Camino alternativo:

- 4.a<durante> El usuario ya está registrado
 - 4.a.1 Sistema informa que el mail de usuario ya está registrado
- 4.b<durante> El mail ingresado no es un mail valido
 - 4.b.1 El sistema informa que no es un mail valido
- 4.c<durante> La contraseña ingresada no es válida
 - 4.c.1 El sistema informa que la contraseña ingresada no es válida
- 5.a<previo> Usuario cancela el registro
 - 5.a.1 Fin de CU.

Postcondición:

Éxito: La cuenta quedó registrada

Éxito alternativo: <vacío>

Fracaso: El usuario no creo la cuenta

Reglas de negocio:

RN010: Las contraseñas de los usuarios deben ser de al menos 8 caracteres y no enteramente numéricas.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Dialogal

Nombre: CUUS03_SingInUsuario

Meta: Realizar el login del usuario

Actores: Usuario

Precondición: El usuario debe estar registrado.

Disparador: Usuario clickea la opción SingIn

Camino básico:

1. El usuario clickea SingIn
2. El sistema solicita un nombre de usuario y contraseña
3. El Usuario ingresa los datos solicitados.
4. El Sistema valida el nombre de usuario y la contraseña ingresada.
5. Usuario confirma el ingreso.
6. Sistema muestra ingreso exitoso.

Camino alternativo:

4.a<durante>El nombre de usuario ingresado no está registrado

4.a.1 El sistema informa que el nombre de usuario ingresado no está registrado

4.b<durante>La contraseña ingresada no es válida

4.b.1 El sistema informa que la contraseña ingresada no es válida

4.c<durante> El usuario olvidó su contraseña

- 4.c.1 Usuario clickea opción “Olvido su cuenta”
- 4.c.2 Sistema solicita mail para la recuperación de contraseña
- 4.c.3 Usuario ingresa el mail de su cuenta
- 4.c.4 Sistema envia mail para la recuperación de la contraseña
- 2.a.<previo> Usuario cancela el ingreso
- 2.a.1 Fin de CU.

Postcondición:

Éxito: Usuario ingresó a su cuenta

Éxito alternativo: Usuario recupero su contraseña

Fracaso: El usuario no ingresó a su cuenta

Reglas de negocio:

RN010: Las contraseñas de los usuarios deben ser de al menos 8 caracteres y no enteramente numéricas.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Dialogal

Nombre: CUUS04_SeguirUsuario

Meta: Seguir un usuario

Actores: Usuario

Precondición: El usuario está registrado y logueado.

Disparador: Usuario ingresa al perfil de otro usuario

Camino básico:

1. El usuario ingresa al perfil de otro usuario
2. Sistema muestra los datos de usuario
3. Usuario selecciona la opción follow
4. Sistema confirma el seguimiento

Camino alternativo:

3.a<durante> El usuario ya seguía al otro usuario

3.a.1 El sistema muestra botón de “Unfollow”

Postcondición:

Éxito: El usuario fue seguido

Éxito alternativo: El usuario actual ya seguía al otro usuario

Fracaso: El usuario no realizó el seguimiento

Reglas de negocio:

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Dialogal

Nombre: CUUS05_AgregarComentario

Meta: Agregar un comentario a una reseña

Actores: Usuario

Precondición: El usuario está registrado y logueado.

Disparador: Usuario ingresa un comentario

Camino básico:

1. El usuario ingresa comentario y clickea enter
2. Sistema registra el comentario y lo muestra

Camino alternativo:

1.a<durante> Usuario cancela el comentario

1.a.1 Fin de CU.

Postcondición:

Éxito: El comentario quedó registrado

Éxito alternativo: <vacío>

Fracaso: El usuario no hizo el comentario

Reglas de negocio:

RN002:El usuario debe estar registrado para poder realizar reseñas y/o comentarios.

Requerimientos Funcionales

RF01 Registrar Usuario.

RF02 Iniciar sesión.

RF03 Cerrar sesión.

RF04 Buscar usuarios.

RF05 Seguir usuarios.

RF06 Realizar reseña sobre álbumes.

RF07: Los usuarios deben poder buscar álbumes utilizando filtros, como género, artista, entre otros.

Requerimientos No Funcionales

RNF01 El sistema debe asegurar que los datos estén protegidos del acceso no autorizado.

RNF02 El sistema debe proporcionar mensajes de error que sean informativos.

RNF03 Se debe dar una confirmación cada vez que se realicen acciones sobre la BBDD.

RNF04. Ponderar reseñas sobre otras teniendo en cuenta distintos factores, como por ejemplo: las lecturas de la reseña, la cantidad de reseñas que realizó el usuario.

Cálculos

Rating: Calcula el puntaje promedio de todas las reseñas (reviews) que hace un usuario.

```
def rating(self):
    # Obtén todos los posts del usuario
    user_posts = Post.objects.filter(author=self)

    # Inicializa variables para el cálculo del promedio
    total_rating = 0
    total_posts = 0

    # Calcula la suma total de los ratings y cuenta los posts
    for post in user_posts:
        total_rating += post.rating
        total_posts += 1

    # Evita la división por cero
    if total_posts > 0:
        # Calcula el promedio
        average_rating = total_rating / total_posts
        return average_rating
    else:
        return 0 # Otra valor predeterminado apropiado si el usuario no tiene
publicaciones
```

Featured_calc: este método se utiliza para calcular un número destacado para una reseña en función de la cantidad de lecturas, la cantidad de publicaciones del autor y el tiempo transcurrido desde la publicación.

La función del método se extiende a través de varios pasos clave:

Ponderación de Factores: El método permite asignar pesos a tres factores esenciales:

- Cantidad de Lecturas (a): Este factor evalúa la popularidad o visibilidad de la reseña en función de la cantidad de veces que ha sido leído o visualizado. Cuanto mayor sea este valor, mayor será el impacto en la evaluación de prioridad al momento de destacar reseñas.
- Cantidad de Publicaciones del Autor (b): Este factor nos permite considerar la experiencia y la contribución del autor de la reseña a MusiBlog. Cuantas más reseñas tenga el autor, mayor será la influencia en el número destacado de la reseña.
- Tiempo Transcurrido desde la Publicación (c): Este factor mide la novedad de la reseña. Cuanto más reciente sea el posteo en comparación con la fecha actual, más significativo será su impacto.

Fecha de Referencia: Para evaluar el factor relacionado con el tiempo, el método establece una "fecha de referencia", que corresponde a la fecha y hora actuales de la aplicación web. Esto se hace para comparar cuántos días han transcurrido desde la publicación del objeto.

Ponderación de factores: Los factores mencionados anteriormente se utilizan en una fórmula ponderada. Cada factor se multiplica por su peso correspondiente y se combinan en un solo número, denominado "featured". Este número sirve como una medida cuantitativa de la relevancia y el impacto de la reseña en cuestión.

Valor Flotante: El resultado del cálculo se convierte en un valor en punto flotante (float) para asegurarse de que el número destacado sea preciso y pueda reflejar adecuadamente las diferencias en la importancia relativa de las reseñas.

```
def featured_calc(self):
    a = 1 # ponderación cantidad de lecturas
    b = 30 # ponderación cantidad de publicaciones del autor
    c = 30 # ponderación cantidad de días desde que se publicó
    fecha_referencia = timezone.now()
    diferencia_dias = (fecha_referencia - self.pub_date).days
```

```
        featured_number = a * self.lecturas + b * self.author.cantidad_publicaciones() -  
c * diferencia_dias  
    return float(featured_number)
```

Stack Tecnológico

Front-end (HTML, CSS y JavaScript en un entorno de Django):

Lenguajes y Tecnologías Principales:

HTML (HyperText Markup Language): En un entorno de Django, HTML se utiliza para definir la estructura de las páginas web y para renderizar datos dinámicos provenientes del backend.

CSS (Cascading Style Sheets): CSS se emplea para definir el aspecto y el diseño de la interfaz de usuario, complementando las plantillas HTML y asegurando una apariencia atractiva y coherente en toda la aplicación.

JavaScript: JavaScript desempeña un papel importante en la interacción del usuario y la mejora de la experiencia en la interfaz de usuario en conjunto con el backend de Django.

Herramientas y Frameworks:

Django Templates: Django ofrece un sistema de plantillas que permite incrustar lógica de backend en las páginas HTML. Esto facilita la presentación de datos dinámicos, como listas de elementos o detalles de objetos, directamente en el HTML.

JavaScript enriquecido por Django: Django puede generar y enviar datos a las vistas utilizando JavaScript. Esto permite una interacción más dinámica entre el front-end y el backend, lo que es particularmente útil para formularios, autenticación y otras características interactivas.

Bibliotecas de JavaScript: Aunque el ejemplo menciona JavaScript Vanilla, en un entorno de Django, es común utilizar bibliotecas de JavaScript, como jQuery o AJAX, para mejorar la eficiencia y la capacidad de respuesta de la aplicación.

Estilo y Diseño:

CSS Personalizado: El estilo de la interfaz de usuario se define mediante CSS personalizado. Esto permite una adaptación precisa de la apariencia de la aplicación a las necesidades específicas del proyecto.

Integración con Estilo de Django: Django facilita la incorporación de CSS y JavaScript en las plantillas HTML a través de etiquetas y filtros personalizados, lo que simplifica la gestión de recursos estáticos.

Back-end (Django):

1. Lenguaje de Programación: Python es el lenguaje de programación principal utilizado en el framework Django.
2. Framework Web: se utiliza Django ya que es un framework web de alto nivel y de código abierto que facilita el desarrollo rápido y seguro de aplicaciones web.
3. Base de Datos: Django admite varias bases de datos, incluyendo PostgreSQL, MySQL, SQLite y Oracle. En nuestro caso se optó por utilizar SQLite.

Link Código fuente

<https://github.com/PeposoElOso/TPSoporte>