

Sistemas de Información.

Universidad de Guanajuato.

Edwin Alfonso Valdez Solis.

28 noviembre, 2019.

Profesor: Juan Carlos Gómez Carranza.

Objetivo.

Hacer el diseño de una base de datos para la compra de un boleto de cine, considerando que el sistema es solo para una sucursal y que alguien necesite ser registrado para comprar el ticket, debes mostrar las películas que hay por día y mostrarlas.

Diagrama entidad - relación.

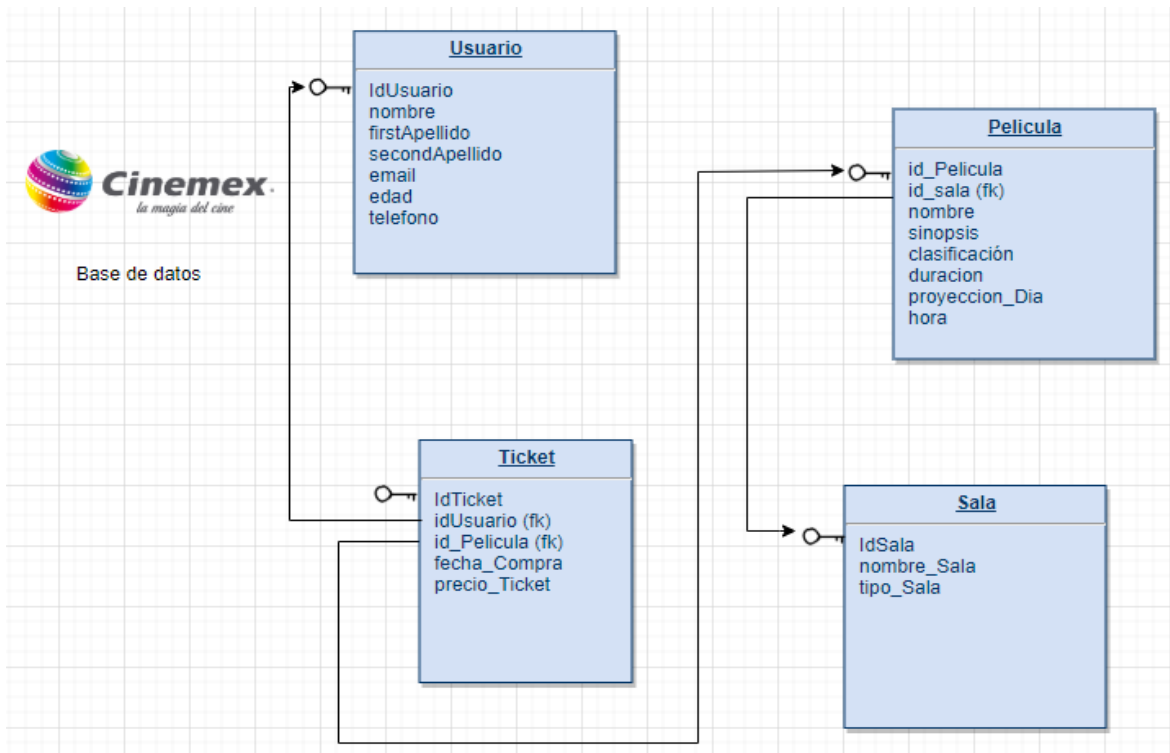


Ilustración 1: Diagrama relacional.

Descripción de la base de datos.

En la base de datos del sistema se encuentran 4 tablas (Usuario, Ticket, Película y Sala), la tabla Ticket tiene dos llaves foráneas las cuales están enlazadas con la película y el usuario ya que un usuario puede comprar un boleto o varios para una película, la película tiene una llave foránea que enlaza con sala ya que una película será proyectada en una sala y no dos películas en la misma sala a la misma hora.

Código.

- **Modelo (Basado en la descripción de la base de datos).**

En Django las tablas se declaran como clases, dichas tablas se generarán en nuestro sistema de bases de datos cuando apliquemos las migraciones.

```
class Usuario(models.Model):
    id_Usuario = models.AutoField(primary_key=True, null=False)
    nombre = models.CharField(max_length=25, null=False, default=False)
    firstApellido = models.CharField(max_length=30, null=False, default=False)
    secondApellido = models.CharField(max_length=20, null=False, default=False)
    email = models.EmailField(max_length=127, null=False)
    edad = models.IntegerField(null=False, default=False)
    telefono = models.CharField(max_length=50, null=False, default=False)

    class Meta:
        verbose_name='Usuario'

    def __str__(self):
        return self.nombre

class Sala(models.Model):
    id_sala = models.AutoField(primary_key=True, null=False)
    SALA = (
        ('1', 'Sala 1'),
        ('2', 'Sala 2'),
        ('3', 'Sala 3'),
        ('4', 'Sala 4'),
        ('5', 'Sala 5'),
    )
    nombre_Sala = models.CharField(max_length=1, choices=SALA, default=False, null=False)
    tipo_Sala = models.CharField(max_length=1, choices=SALA, default=False, null=False)

    class Meta:
        verbose_name='Sala'

    def __str__(self):
        return self.sala
```

```

class Pelicula(models.Model):
    id_Pelicula = models.AutoField(primary_key=True, null=False)
    nombre = models.CharField(max_length=50, null=False)
    sinopsis = models.CharField(max_length=50, null=False)

    CLASIFICACION = (
        ('AA', 'Para todo público'),
        ('A', 'Para todo público'),
        ('B', 'Mayores de 12 años'),
        ('B15', 'Mayores de 15 años'),
        ('C', 'Mayores de edad (+18)'),
    )
    DIA = (
        ('Lunes', 'Lunes'),
        ('Martes', 'Martes'),
        ('Miercoles', 'Miercoles'),
        ('Jueves', 'Jueves'),
        ('Viernes', 'Viernes'),
        ('Sabado', 'Sabado'),
        ('Domingo', 'Domingo'),
    )
    proyeccion_Dia = models.CharField(max_length=10, choices=DIA, null=False)
    Duracion = models.CharField(max_length=127, null=False, default=False)
    id_sala = models.ForeignKey(Sala, on_delete=models.CASCADE, default=False)
    clasificacion = models.CharField(max_length=5, choices=CLASIFICACION, null=False)
    hora = CharField(max_length=127, null=False, default=False)

    class Meta:
        verbose_name = 'Pelicula'

    def __str__(self):
        return self.nombre

class Ticket(models.Model):
    id_Ticket = models.AutoField(primary_key=True, null=False)
    idUsuario = models.ForeignKey(Usuario, on_delete=models.CASCADE)
    id_Pelicula = models.ForeignKey(Pelicula, on_delete=models.CASCADE)
    PRECIO = (
        ('57.00', 'Adulto'),
        ('38.00', 'Niño'),
    )
    precio = models.CharField(max_length=5, choices=PRECIO, default=False, null=False)
    fecha = models.DateTimeField('Fecha de compra', auto_now = True, auto_now_add = False)

```

- Vistas.

```

def buyBoleto(request):
    if request.method == 'POST': #Si la petición es post (mandando datos a
        boleto_form = BoletoForm(request.POST)
        if boleto_form.is_valid():
            boleto_form.save()
            return redirect('index')
        else:
            boleto_form = BoletoForm()
    return render(request, 'cine/buy_boleto.html',{'boleto_form':boleto_for

def pelicula_listar(request):
    pelicula_list = Pelicula.objects.all()
    context = {'pelicula_list':pelicula_list}
    return render(request, 'cine/pelicula_listar.html', context)

```

Vistas.

Los formularios en Django nos permiten escribir menos código, ya que django nos creara las cajas de texto, drop down list, etc. Para ello debemos de definir nuestros formularios en forms.py.

La función buyBoleto verifica si el método es POST, lo que nos mostrara los campos del formulario a llenar, valida el formulario y nos redireccionara a nuestra página principal.

La función película_lista nos lista en la template todas las películas registradas en la base de datos.

Conclusión.

Utilizar Django como framework me facilito la creación de la base de datos, del frontend, el manejo de las direcciones URL, ya que la arquitectura es muy similar a MVC.

En esta práctica pude apreciar el trabajo con una arquitectura de desarrollo de software ya que nos permite implementar buenas prácticas de programación.