EDUCACIÓN
IPSUMTEC
ISSN: 2594 - 2905

SISTEMA WEB PARA LA GESTIÓN DE EVENTOS CON AGENDA DIGITAL USANDO TECNOLOGÍAS OSS CON PYTHON Y VALIDACIÓN DE INFORMES

WEB-BASED EVENT MANAGEMENT SYSTEM WITH A DIGITAL CALENDAR USING PYTHON OSS TECHNOLOGIES AND REPORT VALIDATION

López Aburto María Teresa[1], García López Serbando[2], Gonzalez Guarneros Filiberto[3], Martínez Ramírez Violeta[4], Pérez Irigoyen Miguel Ángel[5]

[1]Tecnológico Nacional de México/Instituto Tecnológico de Puebla, mariateresa.lopez@puebla.tecnm.mx, https://orcid.org/0009-0001-9300-067
[2]Tecnológico Nacional de México/Instituto Tecnológico de Puebla, serbando.garcia@puebla.tecnm.mx, https://orcid.org/0009-0001-9484-9587
[3]Tecnológico Nacional de México/Instituto Tecnológico de Puebla, filiberto.gonzalez@puebla.tecnm.mx, https://orcid.org/0009-0002-2078-7376
[4]Tecnológico Nacional de México/Instituto Tecnológico de Puebla. violeta.martinez@puebla.tecnm.mx, https://orcid.org/0000-0003-1518-786X
[5]Tecnológico Nacional de México/Instituto Tecnológico de Puebla, i17220636.19@puebla.tecnm.mx, https://orcid.org/0009-0003-4246-7290

https://doi.org/10.61117/ipsumtec.v8i2.426

Resumen – El trabajo presenta el desarrollo técnico del "Sistema web para la Gestión y Programación de Eventos del departamento de Actividades Extraescolares del Instituto Tecnológico de Puebla". El propósito del sistema es permitir a los departamentos del Tecnológico solicitar espacios para eventos extraescolares, control de disponibilidad, gestionar los eventos y generar reportes mensuales y trimestrales que incluyan evidencia gráfica.
Se utilizó una arquitectura cliente-servidor basada en tecnologías web modernas como Python con Flask, MariaDB, y el plugin de calendario FullCallendar, además de herramientas como wkhtmltopdf para la generación de reportes.

Palabras Clave-- Flask, Python, FullCallendar wkhtmltopdf, MariaDB.

Abstract -- This paper presents the technical development of the "Web-based System for Event Management and Scheduling for the Extracurricular Activities Department of the Instituto Tecnológico de Puebla." The system's purpose is to allow departments at the Instituto Tecnológico de Puebla to request spaces for extracurricular events, monitor availability, manage events, and generate monthly and quarterly reports that include graphical evidence.
A client-server architecture based on modern web technologies such as Python with Flask, MariaDB, and the FullCallendar calendar plugin was used, as well as tools such as wkhtmltopdf for report generation.

Key words – Flask, Python, FullCallendar wkhtmltopdf, MariaDB.

INTRODUCCIÓN

El desarrollo del sistema se realizó en el contexto del Instituto Tecnológico de Puebla, específicamente dentro del área de la Subdirección de Planeación y Vinculación, en colaboración con el Departamento de Actividades Extraescolares. Esta área es responsable de coordinar, autorizar y calendarizar los eventos extraescolares que involucran a la comunidad tecnológica, tanto a nivel interno como externo.
Durante su desarrollo, se detectó la necesidad de automatizar y optimizar el proceso de gestión de eventos, ya que anteriormente se realizaba mediante un canal en la plataforma Microsoft Teams y formatos físicos. Esto ocasionaba redundancia en la información, retrasos en la aprobación de eventos y poca visibilidad del calendario global.
El entorno laboral estuvo compuesto por un equipo mixto de personal administrativo, docentes encargados de actividades, y residentes de servicio social o residencia profesional. La interacción constante con usuarios reales del sistema (administradores y usuarios solicitantes) permitió obtener retroalimentación continua para mejorar la usabilidad del software desarrollado.
Además, se trabajó bajo un entorno semiescolarizado, lo cual permitió combinar sesiones presenciales y virtuales, así como pruebas en local y en servidores institucionales. Esta flexibilidad fue clave para cumplir con los tiempos establecidos y facilitar las pruebas del sistema antes de su implementación definitiva.

Planteamiento del Problema

En el Instituto Tecnológico de Puebla, el Departamento de Actividades Extraescolares tiene la responsabilidad de coordinar diversos eventos académicos, culturales, deportivos y recreativos que enriquecen la formación integral del estudiantado.

No obstante, la gestión de estos eventos se realiza actualmente de forma manual o mediante métodos no estandarizados, lo cual genera diversos inconvenientes como:

- Dificultad para llevar un control eficiente de las solicitudes de eventos.
- Posibles conflictos en la asignación de fechas y espacios.
- Falta de visibilidad clara sobre los eventos ya programados.
- Retrasos en la atención y respuesta a las solicitudes.

Esta situación puede derivar en desorganización institucional, duplicidad en el uso de espacios y pérdida de tiempo tanto para los usuarios como para el personal administrativo. Por ello, se requiere una solución tecnológica que permita gestionar de manera más eficiente las solicitudes y programación de eventos, así como ofrecer una vista clara y accesible del calendario de actividades extraescolares institucionales.

Objetivo General:

Diseñar e implementar un sistema web para gestionar solicitudes y aprobación de eventos institucionales con generación de reportes PDF.

Objetivos Específicos:

1. Implementación de formulario dinámico para captura de datos de eventos.
2. Crear un panel administrativo para la gestión de las solicitudes de los usuarios.
3. Evitar solapamiento de eventos para evitar conflictos de lugar y fechas.
4. Generación de reportes mensuales y trimestrales de manera dinámica con los eventos divididos por semana y orden cronológico.
5. Carga de imágenes en eventos aprobados para agregar como evidencia visual en los reportes.

Alcances

El sistema desarrollado tiene como propósito principal facilitar la gestión de eventos dentro del Instituto Tecnológico de Puebla, específicamente en el Departamento de Actividades Extraescolares. A continuación, se detallan los alcances principales del sistema:

Permite a los usuarios registrados (docentes y personal administrativo) solicitar eventos a través de un formulario integrado, el cual incluye campos como nombre del evento, fecha de inicio y fin, lugar, descripción, tipo de evento (interno o externo), y responsable.

Los administradores pueden acceder a una vista especializada desde la cual es posible aprobar o rechazar eventos, así como visualizar un resumen de los eventos programados en un calendario interactivo.

El sistema cuenta con un mecanismo de control para evitar duplicidad de eventos en el mismo lugar y horario, bloqueando automáticamente los espacios ya asignados y aprobados.

Una vez aprobado un evento, el usuario podrá subir múltiples imágenes asociadas al mismo, las cuales serán utilizadas para la generación de reportes mensuales y trimestrales en formato PDF.

Se implementó un sistema de roles y autenticación, donde los usuarios solo pueden acceder a la funcionalidad correspondiente según su perfil (usuario o administrador).

El sistema genera automáticamente reportes en PDF, organizando los eventos por semana y agrupando la información relevante, incluyendo imágenes, para facilitar su presentación en instancias institucionales.

El sistema ha sido desarrollado con tecnologías de código abierto: Python, Flask, SQLite (y conexión opcional con MariaDB en servidor), JavaScript (con FullCalendar), HTML/CSS y el motor de plantillas Jinja2.

Puede desplegarse localmente para pruebas y también está preparado para ser subido a un servidor de producción como Render o implementado en XAMPP con conexión a MariaDB.

Python un lenguaje interpretado altamente demandado para desarrollo de aplicaciones web, por su eficiencia, fácil aprendizaje y multiplataforma [1].

El desarrollo web en Python se simplifica gracias a marcos livianos como Flask. La API de Flask permite crear aplicaciones web rápidamente con poco código repetitivo. [2]

FullCalendar visualiza los datos con vista de calendario. Es una excelente opción para mostrar un calendario SharePoint estándar, extrayendo datos de múltiples listas de calendario, datos de listas que no son de calendario o incluso datos externos a SharePoint. [3].

SQLite es un sistema gestor de bases de datos relacionales integrado en múltiples servidores y aplicaciones web, lo que facilita su configuración y conexión con María DB [4].

las plantillas Jinja en Python ofrece una flexibilidad y facilidad de uso, permite que se destaque como una herramienta poderosa para la creación dinámica de contenido en aplicaciones web [5]. Las plantillas suelen componerse solo variables y/o expresiones, las cuales se reemplazan con valores cuando se representa y controlan la lógica en ella. Su sintaxis de está inspirada en Django y Python. [6]

Wkhtmltopdf se autodefine como una herramienta de comandos de código abierto para convertir desde HTML a un documento PDF [7].

Estado del Arte

Existe estudios difundidos en internet sobre la mancuerna que puede hacerse con aplicativos webs del lado del servidor, quienes combinan framesworks para Python, entre los más populares se encuentran Pyramid, Flask, Bottle. Se afirma que Python junto a Django optimizan el tiempo invertido en una producción backend del aplicativo, ya que la estructura de Django proporciona propiedades de seguridad más confiables. Esto disminuye significativo el costo de la inversión en las instituciones públicas su desarrollo tecnológico.

En Colombia, el trabajo publicado “Desarrollo de una interfaz web basada en Python para el análisis acústico del habla en tiempo real”, expone la facilidad de crear una aplicación Web basado en las herramientas de Python exitosamente en el análisis de audio en tiempo real y de manera offline. La interfaz diseñada con wireframes, permitió identificar el flujo de interacción del usuario y

un procesamiento backend estructurado gracias a Flask y Parselmouth para el análisis de audio. [8].

El proyecto "Diseño e implementación de un sistema administrador de reportes de novedades de empleados basado en una aplicación web" desarrollado en Colombia también eligió a Python como lenguaje principal para la creación del sistema que gestiona proceso de reportes resultados de licencias, incapacidades, retiros, traslados de empleados dentro de una empresa. Se obtuvo una excelente automatización para una mejor toma de decisiones. [9]

En Ecuador, el sistema de control académico desarrollado en un ambiente Web fue basado en Python con Bootstrap4 y el framework Flask, se hizo uso de plantillas HTML con las que se obtuvo una interfaz amigable y dinámica que facilitó la interacción web. Los resultados fueron publicados con el nombre del trabajo "Sistema de Control Académico Basado en Python Y Cloud Computing, a Nivel de Secundaria en la Ciudad de Santo Domingo" [10].

El trabajo llamado "Sistema web basado en reglas de asociación para apoyar en el proceso de ventas de la empresa Ayala Motors, 2022", muestra una implementación del sistema web *recomendador*, donde fueron utilizados herramientas como el Google *colab*, Python, Visual studio code, PHP, Laravel, PostgreSQL con las cuales, fue posible acercar una herramienta tecnológica que apoyará eficazmente a empleados en el proceso de ventas dentro de la empresa [11].

DESARROLLO

El sistema fue desarrollado localmente sobre el siguiente entorno:

- Lenguaje Principal: Python
- Framework: Flask
- Base de datos: MariaDB (XAMPP)
- Frontend: HTML5, CSS3 (personalizado) + FullCallendar
- Reportes PDF: wkhtmltopdf
- IDE: Visual Studio Code
- Servidor Final: Windows Server

Metodología de desarrollo

El enfoque ágil permitió iterar rápidamente sobre el desarrollo de funcionalidades clave como el calendario de eventos, el registro y autenticación de usuarios, la gestión de solicitudes, la aprobación/rechazo por parte del administrador y la generación de reportes mensuales y trimestrales.

Cada iteración incluía:

- Revisión de requerimientos con la asesora.
- Implementación de una funcionalidad específica.
- Pruebas y retroalimentación inmediata.
- Ajustes con base en la retroalimentación recibida.

Tecnologías utilizadas

- Back-end: Framework Flask (Python), con estructura modular basada en rutas, modelos y plantillas. Se usó Flask-Login para la autenticación y control de sesiones.
- Front-end: HTML5, CSS3 y JavaScript, con apoyo de la librería FullCalendar.js para el calendario dinámico e interactivo.
- Motor de base de datos: MariaDB para entornos de producción.
- Plantillas dinámicas: Se utilizó Jinja2 para construir vistas dinámicas y personalizadas.
- Generación de reportes en PDF: Se empleó PDFKit junto con wkhtmltopdf para convertir plantillas HTML en archivos PDF de forma automática.
- Entorno de desarrollo: Visual Studio Code en Windows 10, con entorno virtual Python (venv) y ejecución local con Flask. El sistema fue preparado para su despliegue en un Windows Server con XAMPP.

Código de la creación y gestión de eventos

solicitar_evento: Esta función permite registrar eventos nuevos desde el formulario de los usuarios o el admin. El evento queda pendiente de aprobación hasta ser revisado por un admin para su futura aprobación o negación para corrección:

Imagen 1. *Solicitar de eventos*.

```
@eventos.route('/solicitar_evento', methods=['POST'])
@login_required
def solicitar_evento():
    nombre = request.form['nombre']
    lugar = request.form['lugar']
    responsable = request.form['responsable']
    descripcion = request.form['descripcion']
    fecha_inicio = datetime.strptime(request.form['fecha_inicio'], '%Y-%m-%dT%H:%M')
    fecha_fin = datetime.strptime(request.form['fecha_fin'], '%Y-%m-%dT%H:%M')

    participantes = request.form.get('participantes')
    genero = request.form.get('genero')
    tipo_evento = request.form.get('tipo_evento')

    if tipo_evento == 'Interno':
        organizacion = request.form.get('organizacion')  # de la lista
    elif tipo_evento == 'Externo':
        organizacion = request.form.get('organizacion_texto')  # texto libre
    else:
        organizacion = None

    if fecha_fin < fecha_inicio:
        flash("La fecha y hora de fin no puede ser anterior a la de inicio.", "danger")
        return redirect(url_for('eventos.user_index'))
```

Fuente. *Elaboración propia* (2025).

Parte 1 del bloque solicitar_eventos donde se envían los datos ingresados en el formulario y revisa que las fechas sean coherentes. Ver imagen 1.

Parte 2 de solicitar_eventos donde antes de guardar el evento en la base de datos se revisa si hay un evento ya aprobado con el mismo lugar y rango de fechas. Ver imagen 2.

Imagen 2. *solicitar_eventos previa validación*.

ISSN: 2594 - 2905

Fuente. *Elaboración propia* (2025).

Parte 3 de solicitar_evento donde muestra al usuario que su evento fue enviado al admin de manera correcta para su aprobación o rechazo. Ver imagen 3.

Imagen 3. *Solicitar evento se apreiba o rechaza*.

Fuente. *Elaboración propia* (2025).

Código administración de eventos

Bloque "aprobar", el admin al aprobar un evento lo envía a la base de datos para ser mostrado en el calendario y permitir la subida de imágenes para los reportes en PD. Ver imagen 4.

Imagen 4. *Valida y muestra calendario*.

Fuente. *Elaboración propia* (2025).

Bloque "rechazar", al rechazar un evento el admin debe ingresar un motivo el cual será visualizado por el usuario para hacer correcciones de ser necesario para volver a enviarlo. Ver imagen 5.

Imagen 5. *Rechazar eventos*.

Fuente. *Elaboración propia* (2025).

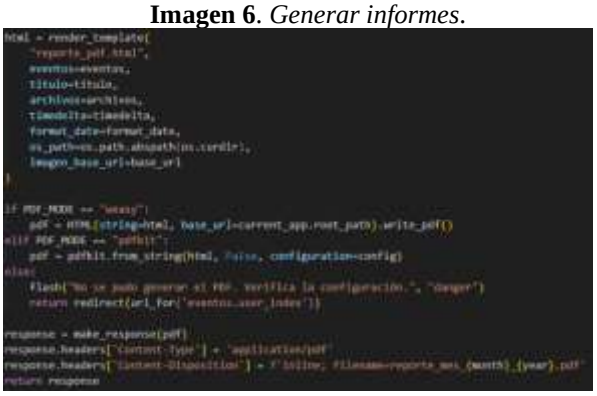
Código de la generación de reportes

En el siguiente bloque de código se presenta como el sistema genera los reportes mensuales y trimestrales:

Parte 1 del bloque reporte_mes donde se revisa que mes se está viendo en el calendario al momento de generar el prf, así como revisar si hay eventos en el mes, la generación del título y la revisión de si hay imágenes en la carpeta para considerarlas en el ciclo de generación

Parte 2 del bloque reporte_mes donde se procesa la información para generar el PDF. Ver imagen 6.

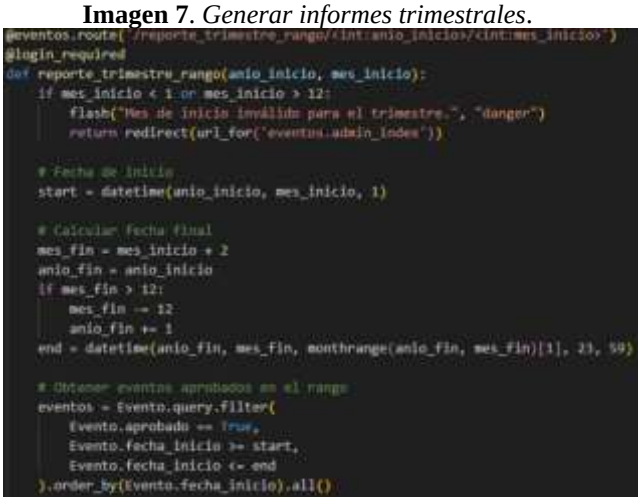
Imagen 6. *Generar informes*.



Fuente. *Elaboración propia* (2025).

Parte 1 del bloque reporte_trimestre_rango donde se calcula la fecha inicial del reporte en base al mes que se está viendo en el calendario, así como la obtención de eventos. Ver imagen 7.

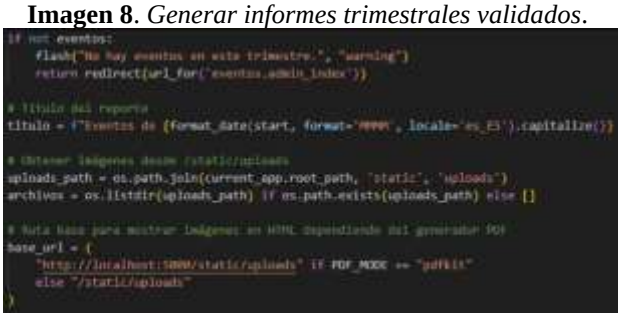
Imagen 7. *Generar informes trimestrales*.



Fuente. *Elaboración propia* (2025).

Parte 2 del del bloque reporte_trimestre_rango donde se revisa si no hay eventos en el mes, se genera el título del reporte y se verifican las imágenes para considerarlas en la generación. Ver imagen 8.

Imagen 8. *Generar informes trimestrales validados*.



Fuente. *Elaboración propia* (2025).

Parte 3 del bloque reporte_trimestre_rango donde se asigna toda la información obtenida para el reporte, así como verificar si el generador wkhtmltopdf existe. Ver imagen 9.

Imagen 9. *Generar informes por rango*.

```
html = render_template(
    "reporte_pdf.html",
    eventos=eventos,
    titulo=titulo,
    archivos=archivos,
    timedelta=timedelta,
    format_date=format_date,
    os_path=os.path.abspath(os.curdir),
    es_trimestral=True,
    imagen_base_url=base_url
)

# Generar el PDF
if PDF_MODE == "weasy":
    pdf = HTML(string=html, base_url=current_app.root_path).write_pdf()
elif PDF_MODE == "pdfkit":
    options = {
        'enable-local-file-access': None
    }
    pdf = pdfkit.from_string(html, False, configuration=config, options=options)
else:
    flash("No se pudo generar el PDF. Verifica la configuración.", "danger")
    return redirect(url_for('eventos.admin_index'))
```

Fuente. *Elaboración propia* (2025).

Parte 4 del bloque reporte_trimestre_rango donde se genera el reporte para su visualización y descarga

Imagen 10. *Descarga del informe en PDF*.

```
# Devolver el PDF como respuesta
response = make_response(pdf)
response.headers['Content-Type'] = 'application/pdf'
response.headers['Content-Disposition'] = f'inline;
return response
```

Fuente. *Elaboración propia* (2025).

Código de la subida de imágenes

A continuación, el código donde se procesan las imágenes a subir, se aceptan diferentes formatos y se les asigna un nombre según el evento:

Parte 1 del bloque subir_imagenes donde se procesan las imágenes para saber si son válidas, así como la asignación de nombre según el nombre del evento.

Imagen 11. *Proceso cargar imágenes*.

```
@eventos.route('/subir_imagenes/<int:evento_id>', methods=['POST'])
@login_required
def subir_imagenes(evento_id):
    evento = Evento.query.get_or_404(evento_id)

    if evento.usuario_id != current_user.id or not evento.aprobado:
        flash("No puedes subir imágenes para este evento.", "danger")
        return redirect(url_for('eventos.user_index'))

    archivos = request.files.getlist('imagenes')
    if not archivos or archivos[0].filename == '':
        flash("No se seleccionó ninguna imagen.", "warning")
        return redirect(url_for('eventos.user_index'))

    extensiones_permitidas = {'png', 'jpg', 'jpeg', 'gif', 'webp'}
    rutas_guardadas = []

    uploads_dir = os.path.join(current_app.root_path, 'static', 'uploads')
    os.makedirs(uploads_dir, exist_ok=True)

    nombre_base = limpiar_nombre(evento.nombre)

    # Buscar cuántas imágenes ya existen con este nombre base
    archivos_existentes = [
        f for f in os.listdir(uploads_dir)
        if f.startswith(f"evento_{nombre_base}_")
    ]
```

Fuente. *Elaboración propia* (2025).

Parte 2 del bloque subir_imagenes, en esta parte se enumeran la o las imágenes según se revisa si ya hay una anteriormente y luego las sube a la carpeta uploads. Ver imagen 11.

Imagen 11. *Proceso subida de imágenes*.

```
contador_inicial = len(archivos_existentes) + 1

for i, archivo in enumerate(archivos, start=contador_inicial):
    if archivo and '.' in archivo.filename:
        extension = archivo.filename.rsplit('.', 1)[-1].lower()
        if extension not in extensiones_permitidas:
            continue

        nombre_archivo = f"evento_{nombre_base}_{i}.{extension}"
        ruta_completa = os.path.join(uploads_dir, nombre_archivo)
        archivo.save(ruta_completa)

        rutas_guardadas.append(f"static/uploads/{nombre_archivo}")

if rutas_guardadas:
    evento.imagen = rutas_guardadas[-1]  # Última subida como principal
    db.session.commit()
    flash(f"Se subieron {len(rutas_guardadas)} imagen(es) correctamente.", "success")
else:
    flash("No se subió ninguna imagen válida.", "warning")

return redirect(url_for('eventos.user_index'))
```

Fuente. *Elaboración propia* (2025).

Pantalla principal:

Esta pantalla incluye la sección de inicio de sesión para usuarios y administradores, así como el calendario donde se visualizan todos los eventos solicitados por los usuarios y aprobados por el administrador. Pantalla Principal del sistema, se visualizan los logos del instituto, la sección de inicio de sesión y el calendario con los eventos. Ver imagen 12.

Imagen 12. *Proceso cargar de imágenes*.

Fuente. *Elaboración propia* (2025).

Vista principal del administrador

En esta pantalla el administrador puede ver el calendario de eventos, aprobar solicitudes, eliminar o rechazar eventos, y generar reportes mensuales y trimestrales:

Parte 1 de la pantalla del administrador, incluye el calendario, botones para la generación dinámica de los reportes y un botón para borrar todos los eventos que haya en el mes que se esté visualizando. Ver imagen 13.

Imagen 13. Interfaz con componentes y calendario.

Fuente. *Elaboración propia* (2025).

Parte 2 de la pantalla del administrador, se puede visualizar una sección con los eventos pendientes pedidos por los usuarios o hechos por el mismo administrador, junto se encuentran los botones de aceptar o rechazar los eventos, debajo se encuentra un formulario que permite al administrador realizar sus propios eventos de ser necesario. Ver imagen 14.

Imagen 14. *Interfaz con componentes y calendario*.

Fuente. *Elaboración propia* (2025).

Vista del usuario

Los usuarios regulares pueden solicitar eventos mediante un formulario y consultar el estado de sus eventos:
Parte 1 de la pantalla de usuarios, se puede visualizar un botón de cerrar sesión y el calendario con los eventos.

Imagen 15. *Calendario de eventos y salida*.

Fuente. *Elaboración propia* (2025).

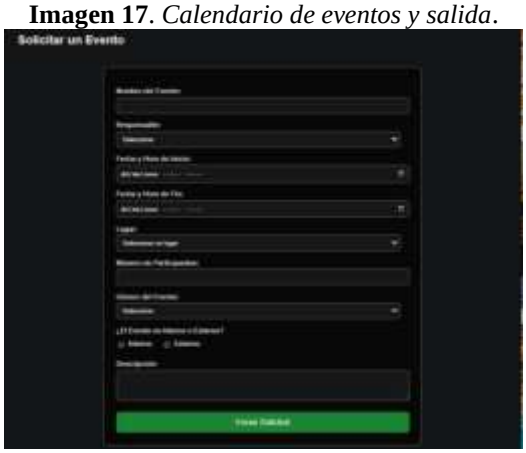
Parte 2 de la pantalla de usuarios, sección de eventos solicitados por el usuario, cuenta con los botones de edición para corregir errores en la información o un botón de eliminación para quitar el evento si es necesario. Ver imagen 16.

Imagen 16. *Calendario de eventos y salida*.

Fuente. *Elaboración propia* (2025).

Parte 3 de la pantalla de usuarios, un formulario para la creación de eventos cuenta con la información importante como el nombre del evento, fechas, lugar, participantes o si el evento fue solicitado interna o externamente, así como una sección de descripción para aclaraciones. Ver imagen 17.

Imagen 17. *Calendario de eventos y salida*.



Fuente. *Elaboración propia* (2025).

DISCUSIÓN Y ANÁLISIS DE RESULTADOS

Generación de reportes

El sistema permite generar reportes tanto mensuales como trimestrales en formato PDF, estos reportes son temporales y son eliminados cuando se regresa a la pantalla anterior para evitar que la memoria del sistema se sature, para conservar el contenido el PDF debe ser descargado:
PDF de reporte mensual, se incluye el titulo con el mes y cada evento está en orden cronológico divididos por una línea para dar claridad visual. Ver imagen 18.

Imagen 18. *Informe mensual*.

Fuente. *Elaboración propia* (2025).

Visualización de datos de eventos del calendario

Al hacer clic en un evento del calendario se abre una ventana lateral a la derecha del navegador la cual contiene la información detallada del evento o eventos del día seleccionado: Al poner la flecha del ratón en un evento se puede visualizar la información individual de un evento. Ver imagen 19.

Imagen 19. *Detalle del evento*.

Fuente. *Elaboración propia* (2025).

Al hacer clic cualquier evento se desplegará una ventana lateral la cual tiene toda la información del evento, esta ventana puede ser cerrada simplemente haciendo clic fuera de ella o con el icono "X" arriba a la derecha del panel. Ver imagen 20.

Imagen 20. *Detalle del evento*.

Fuente. *Elaboración propia* (2025).

Resultados obtenidos

El resultado del proyecto es un sistema funcional, intuitivo y ligero, que logra digitalizar y automatizar el proceso de registro, control y documentación de eventos institucionales, lo cual anteriormente se realizaba de forma manual o mediante formularios básicos.

Principales logros y beneficios del sistema

- Interfaz clara y accesible, que permite consultar eventos de manera mensual e interactiva con solo hacer clic sobre una fecha.
- Diferenciación efectiva de roles, permitiendo personalizar la experiencia del usuario y del administrador desde el inicio de sesión.
- Validación automática de solicitudes, previniendo duplicaciones de lugar o conflicto de horarios.
- Bloqueo inteligente de fechas y lugares ya ocupados por eventos aprobados.
- Estados claros de cada solicitud: aprobado, rechazado (con motivo visible), editable o eliminable en caso de rechazo.
- Carga y gestión de imágenes por evento, para evidenciar actividades y respaldarlas documentalmente.
- Reportes PDF automáticos, organizados por semana, con integración de imágenes y datos útiles para archivo o presentación institucional.
- Desempeño estable en entorno local, con buena velocidad de respuesta y bajo nivel de dificultad para usuarios nuevos.
- Escalabilidad del sistema, preparado para funcionar en servidores como XAMPP o en la nube, gracias a su arquitectura modular y uso de tecnologías ampliamente compatibles (Tabla 1).

Tabla 1. *Datos comparativos*.

Criterio	Proceso anterior (Canal de Teams + oficina de coordinación)	Proceso actual (Sistema web)
Medio de solicitud	Mensajes en un canal de Microsoft Teams y entrega física de formatos.	Formulario en línea integrado en la página web del sistema.
Tiempo de aprobación	Variable; dependía de la disponibilidad del personal y la revisión manual.	Inmediato en la recepción; validación automática de datos y disponibilidad.
Control de agenda	Manual, con riesgo de duplicación de lugares y fechas.	Automático, con bloqueo de lugares y fechas ya ocupadas.
Visibilidad de eventos	Limitada; requería consultar a la coordinación o el canal de Teams.	Total; calendario interactivo accesible para usuarios y administradores.
Gestión de cambios	Requería comunicación directa con la coordinación y posible reenvío de formatos.	Edición en línea de eventos no aprobados y reenvío automático para revisión.
Registro de evidencias	No existía un método centralizado; dependía de envío por correo o mensajería.	Carga de imágenes directamente en el evento aprobado para reportes.
Generación de reportes	Manual, utilizando	Automática en PDF, mensual o

	documentos de texto o presentaciones.	trimestral, con imágenes y datos consolidados.
Acceso y seguridad	Sin control de roles; cualquier persona en el canal podía ver solicitudes.	Control de acceso por roles (usuario y administrador) con autenticación.

Fuente. *Elaboración propia* (2025).

Pruebas funcionales

Durante todo el proceso de desarrollo se realizaron pruebas funcionales continuas, enfocadas en cada módulo individual y en la integración general del sistema.

Pruebas realizadas

- Simulación de distintos tipos de usuarios y pruebas de autenticación.
- Verificación de campos del formulario, fechas válidas y detección de conflictos de lugar.
- Revisión del flujo completo de aprobación y rechazo de eventos.
- Pruebas de comportamiento dinámico de la interfaz (por ejemplo, mostrar/ocultar botones según el estado del evento).
- Generación de reportes y revisión detallada del contenido, formato y calidad de las imágenes insertadas.

Estas pruebas permitieron detectar errores menores en etapas tempranas, los cuales fueron corregidos en cada ciclo de iteración. Al finalizar el desarrollo, el sistema demostró una operación estable en todas las funciones implementadas, cumpliendo con los requerimientos establecidos desde el inicio del proyecto.

Discusión

Con tecnología en la nube de Amazon (AWS), su expertis permite afirmar que Python ofrece gran facilidad para implementación de sistemas multiplataforma a través de diferentes sistemas operativos de computadora como Windows, macOS, Linux y Unix; además de la fuerte protección de datos en el envío de información en la red gracias funciones complejas con las que cuenta para su desarrollo web. [1]

Contar con una agenda digital incluida en un sistema web, permite al usuario acceder con facilidad en cualquier parte del mundo, gestionar eventos de importancia para los procesos de uso de tiempos y espacios sin empalmes. La coordinación de horarios es fundamental para todos los usuarios concurrentes [13]

John Hill [14] describe “Los calendarios son fundamentales para documentar y realizar un seguimiento de las fechas y eventos importantes de nuestras vidas.”, destacando que una gestión de proyectos es esencial tanto para los usuarios como, los involucrados en el proceso administrativo estén actualizados del estado que guarda cada evento.

Al obtener automáticamente los informes digitales, se reduce hasta un 30% de tiempo y reducción total de papel. Además, la información queda centralizada y lista para su interpretación favoreciendo la toma de decisiones informada y por consecuencia la comunicación se torna más fluida e inmediata institucionalmente. Los datos permanecen disponibles y recuperables en tiempo real, y de esta forma es mucho más fácil asegurarse la integridad de la información a lo largo del proceso. [15]

CONCLUSIONES

El desarrollo del sistema de gestión de eventos para el Departamento de Actividades Extraescolares del Instituto Tecnológico de Puebla representó una experiencia integral en el ciclo completo del desarrollo de software. A lo largo del proyecto se aplicaron buenas prácticas de ingeniería de software, desde el análisis del problema y la definición de requerimientos, hasta la implementación y validación funcional con usuarios reales.

Una de las principales fortalezas del sistema es su enfoque centrado en el usuario, permitiendo que tanto personal administrativo como usuarios generales gestionen eventos de forma clara y segura. El uso de Flask como framework principal facilitó una arquitectura modular y eficiente, mientras que tecnologías como SQLite/MariaDB, HTML/CSS/JS, FullCalendar y PDFKit hicieron posible una interfaz intuitiva y funcionalidades robustas como la carga de múltiples imágenes, generación de reportes automáticos y control de estados de eventos.

Asimismo, el despliegue en un entorno controlado y la compatibilidad con plataformas como XAMPP (MariaDB) y Render para producción, garantizan la portabilidad y adaptabilidad del sistema ante futuros escenarios reales.

Durante el desarrollo se presentaron varios retos, como el manejo de rutas de archivos estáticos en distintos entornos, la validación de datos desde el front-end y back-end, y la lógica para mantener integridad en las solicitudes simultáneas de eventos. Todos ellos fueron abordados exitosamente aplicando una metodología incremental basada en la filosofía ágil, validando constantemente avances con la asesora del proyecto.

Finalmente, el sistema no solo cumple con los objetivos funcionales planteados al inicio, sino que también queda preparado para una futura escalabilidad, como la generación de reportes trimestrales, integración con sistemas de autenticación institucional y mejoras visuales. La experiencia adquirida en este proyecto fortalece las capacidades del desarrollador tanto a nivel técnico como organizacional, sentando las bases para futuros trabajos de mayor complejidad.

Trabajo futuro

Una línea de investigación futura podría centrarse en el desarrollo de sistemas avanzados de generación de informes interactivos que integren visualización de datos dinámica, análisis predictivo y herramientas de apoyo a la toma de decisiones. Este enfoque permitiría estudiar cómo el uso de gráficos, filtros inteligentes e infografías automatizadas contribuye a mejorar la precisión de las

proyecciones institucionales y optimiza la gestión de recursos [16].

Otra línea de investigación podría orientarse al estudio del uso de Python como lenguaje principal para el análisis de datos aplicado al desarrollo web. La investigación podría abordar la integración de bibliotecas como Pandas, NumPy y TensorFlow en entornos de servidor para construir aplicaciones capaces de procesar y visualizar grandes volúmenes de información en tiempo real [17].

BIBLIOGRAFÍA

[1] Amazon Web Services. ¿Qué es Python? [Internet]. Amazon; 2024 [citado 27 ago 2025]. Disponible en: https://aws.amazon.com/es/what-is/python/

[2] Ramotion. Python para desarrollo web [Internet]. 2025 [publicado 22 jul 2025, citado 27 ago 2025]. Disponible en: https://www.ramotion.com/blog/web-development-in-python/#:~:text=Simplicidad%20y%20facilidad%20de%20uso,con%20un%20m%C3%ADnimo%20de%20c%C3%B3digo.

[3] Microsoft Learn. Migrar la solución de jQuery y FullCalendar compilada con el elemento web Editor de scripts a SharePoint Framework [Internet]. 2022 [publicado 29 jun 2022, citado 27 ago 2025]. Disponible en: https://learn.microsoft.com/es-es/sharepoint/dev/spfx/web-parts/guidance/migrate-jquery-fullcalendar-script-to-spfx

[4] Hassan A. ¿Qué es SQLite? [Internet]. BUILTIN; 2025 [publicado 24 abr 2025, citado 27 ago 2025]. Disponible en: https://builtin.com/data-science/sqlite#:~:text=Beneficios%20de%20SQLite,c%C3%B3digo%20aparte%20en%20otro%20lenguaje.

[5] Atareao. El poder de Jinja y Python [Internet]. 2024 [publicado 9 abr 2024, citado 27 ago 2025]. Disponible en: https://atareao.es/podcast/el-poder-de-jinja-y-python/#:~:text=Sobre%20las%20ventajas%20de%20usar,necesidades%20espec%C3%ADficas%20de%20sus%20proyectos.

[6] Jinja Project. Documentación del diseñador de plantillas [Internet]. 2007 [citado 27 ago 2025]. Disponible en: https://jinja.palletsprojects.com/en/stable/templates/

[7] WKHTMLTOPDF. ¿Qué es? [Internet]. [s.f.] [citado 27 ago 2025]. Disponible en: https://wkhtmltopdf.org/

[8] Molina Giraldo MA, Oñate González AF. Desarrollo de una interfaz web basada en Python para el análisis acústico del habla en tiempo real [Internet]. Bogotá: Pontificia Universidad Javeriana; 2024 [citado 27 ago 2025]. Disponible en: https://repository.javeriana.edu.co/items/227d28c0-c814-4348-b895-8402d8e30972

[9] López Bran DA, Montoya Sáenz A. Diseño e implementación de un sistema administrador de reportes de novedades de empleados basado en una aplicación web [Internet]. Rev Univ Catol Oriente. 2020;31(45):28-44 [citado 28 ago 2025]. Disponible en: https://revistas.uco.edu.co/index.php/uco/article/view/281

[10] Tejada Campos DA. Sistema de control académico basado en Python y cloud computing, a nivel de secundaria en la ciudad de Santo Domingo [Internet]. Santo Domingo (Ecuador): Universidad Regional Autónoma de los Andes; 2020 [citado 28 ago 2025]. Disponible en: https://dspace.uniandes.edu.ec/handle/123456789/11358

[11] Onofre Ávila BA. Análisis de aplicaciones web utilizando Python para el desarrollo del lado del Backend en instituciones públicas [Internet]. Babahoyo (Ecuador): Universidad Técnica de Babahoyo; 2021 [citado 28 ago 2025]. Disponible en: https://dspace.utb.edu.ec/handle/49000/9534

[12] de Iman UA, Lila N. Sistema web basado en reglas de asociación para apoyar en el proceso de ventas de la empresa Ayala Motors [Internet]. Chiclayo (Perú): Universidad Católica Santo Toribio de Mogrovejo; 2022 [citado 28 ago 2025]. Disponible en: http://hdl.handle.net/20.500.12423/7815

[13] Doodle. Ventajas de una agenda digital: ahorro de tiempo en el trabajo [Internet]. 2025 [publicado 24 mar 2025, citado 7 sep 2025]. Disponible en: https://doodle.com/es/benefits-of-a-digital-calendar-saving-time-in-the-workplace/

[14] Hill J. Las ventajas de tener un calendario digital frente a un calendario de papel [Internet]. Calendar.com; 2023 [publicado 16 mar 2023, citado 7 sep 2025]. Disponible en: https://www.calendar.com/blog/the-advantages-of-having-a-digital-calendar-over-a-paper-calendar/#:~:text=Las%20citas%20se%20pueden%20mover,como%20si%20siempre%20hubiera%20existido.

[15] Kizeo Forms. Informes digitales: usos y beneficios para tu empresa [Internet]. [s.f.] [citado 7 sep 2025]. Disponible en: https://www.kizeo-forms.com/es/blog/informes-digitales-usos-y-beneficios-para-tu-empresa/#:~:text=Ventajas%20de%20los%20informes%20digitales,tiempos%20hasta%20en%20un%2030%25.

[16] Duek C. Los reportes digitales son una tendencia a nivel mundial [Internet]. Kaleida Digital; 2024 [publicado 23 jun 2020, citado 7 sep 2025]. Disponible en: https://kaleidadigital.com/es/los-reportes-digitales-son-una-tendencia-a-nivel-mundial/#:~:text=Los%20beneficios%20de%20los%20reportes,a%20las%20empresas%20y%20organizaciones.

[7] Brainvire Infotech Inc (2025). Explorando el papel de liderazgo de Python en el análisis de datos y sus ventajas clave. [Internet] [publicado el 10 de septiembre de 2024, consultado 9 de noviembre de 2025]. Disponible en: https://www.brainvire.com/blog/python-data-analytics-innovation/#:~:text=de%20otros%20lenguajes.-,Rendimiento%20y%20escalabilidad,el%20an%C3%A1lisis%20de%20big%20data.

Tabla de Roles de Contribución

Rol	Autor (es)
Conceptualización	Miguel Ángel Pérez Irigoyen

Curación de datos	Miguel Ángel Pérez Irigoyen
Metodología	Filiberto Gonzalez Guarneros
Administración del proyecto	María Teresa López Aburto
Recursos	Revisión y edición – Violeta Martínez Ramírez
Software	Miguel Ángel Pérez Irigoyen
Supervisión	Serbando García López
Validación	Miguel Ángel Pérez Irigoyen
Visualización	María Teresa López Aburto
Redacción	Borrador original – Violeta Martínez Ramírez
Redacción	Revisión y edición – Violeta Martínez Ramírez

.