

Taller de Lenguajes de Programación III.

Trabajo Práctico Integrador.

1. Objetivos

- Integración de los conocimientos adquiridos durante el desarrollo de las unidades temáticas del módulo y de otras asignaturas, en el proceso de elaboración de un sistema.
- Estimular la expresión oral y escrita.
- Presentar las distintas etapas del ciclo de vida de una aplicación web aplicadas al desarrollo de un sistema particular.
- Utilizar la presentación en etapas como estrategia de motivación y aprendizaje.
- Estimular la producción de presentaciones formales.

2. Contenidos

- Marco teórico
Para la redacción del mismo es importante tener en cuenta la aplicación de los siguientes conceptos al proyecto seleccionado por el grupo.
- Entorno web (servidores, lenguajes de programación, navegadores web).
- Arquitectura cliente – servidor.
- Herramientas utilizadas en cada etapa del ciclo de vida de desarrollo de una aplicación web, por ejemplo, diagrama de Gantt, mapa de sitio, frameworks, entre otros.
- Diseño, diagramación y normalización de la base de datos.
- Breve descripción de los lenguajes de programación web.
- Explicación de los componentes del framework Flex Mind empleados.
- Pruebas de verificación y validación para corroborar el funcionamiento

Actividades

Se solicita que los estudiantes desarrollen un sistema teniendo en cuenta las siguientes pautas:

1. Investigar otros aplicativos similares para aclarar y definir los objetivos, recursos y expectativas en torno a la aplicación.
2. Identificar las entidades que intervienen en el sistema.
3. Establecer el plan de acción a seguir durante el proceso de desarrollo, considerando los siguientes requisitos:

Opción C: Guía turística.

- El sistema debe contar con una sección de puntos históricos, paseos culturales, comidas típicas, centros de salud, oficinas gubernamentales y especies autóctonas.

- Cada una de estas secciones debe contener una grilla, un formulario de carga y las validaciones correspondientes para cada acción.
 - Se debe cargar datos por departamento y localidad para cada una de las secciones así se podrán aplicar diversos filtros.
 - Se debe contar con un flexReport que deberá basarse en los filtros seleccionados para mostrar los registros. Por ejemplo, si se selecciona puntos históricos de la ciudad de Formosa, se visualizarán todos los puntos históricos con sus correspondientes características considerando las localidades de la provincia.
 - Adicionalmente, el formulario permitirá aplicar filtros seleccionando una de las secciones y uno de los datos cargados para dicha sección, es decir, si busco un lugar en el departamento Pirané donde se cocine Sopa paraguaya, que me liste todos los lugares en un flex Combo, por ejemplo, y me permita seleccionar uno que posteriormente se visualizará en el reporte.
4. Seleccionar la estructura y el contenido de la aplicación, considerando las necesidades a satisfacer y los intereses del público al que va dirigido.
 5. Diseñar, diagramar y crear la base de datos que se emplea.
 6. Codificar utilizando el framework Flex Mind. En el desarrollo del trabajo escrito, agregar una breve explicación de la lógica utilizada para la programación del proyecto.
 7. Realizar diversas pruebas para asegurar el funcionamiento del sistema en diversos navegadores, verificando la lógica aplicada. Dejar constancia de cada una de las pruebas ejecutadas.
 8. Redactar conclusiones que surjan del trabajo realizado.
 9. Presentar el trabajo integrador en formato PDF con nombre y apellido. Subirlo al aula virtual.
 10. Preparar una presentación para exponer el trabajo en clase.

Desarrollo:

Marco Teórico:

Herramientas utilizadas:

- **Flexide:**
Este sistema es el encargado de gestionar los códigos fuente de APPS web, brindando una interfaz online que facilita la programación para el framework Flex Mind. Además nos ofrece múltiples ventajas tales como proteger el código fuente de múltiples accesos, gestión de bases de datos de mariaDB, auditar acciones mediante reportes, control de versiones, etc.

- **Flex Mind:**

Es un framework basado en componentes que crean estructuras en tiempo de ejecución, cuyo objetivo es facilitar y agilizar la programación de sistemas mediante una interfaz amigable y sencilla que permita la generación y actualización de módulos..

Está basado en el uso y desarrollo de componentes independientes y vinculados entre sí, siendo cada componente un objeto de javascript que se ejecuta para obtener el resultado esperado según el componente usado.

Cada componente es configurado a través de archivos php y html, esto mediante la librería templates), que ejecutan funciones javascript (que crean un objeto).

Uno de los requisitos fundamentales para el correcto funcionamiento del framework es un buen diseño y normalización de base de datos, respetando los estándares de nombres y nomenclaturas, gracias a estas imposiciones, se pueden disminuir en gran medida los errores.

Otro aspecto clave del cual depende en gran medida es la estructura de directorios, todos los módulos se ubican en el directorio “módulos” y es aquí donde las subcarpetas correspondientes a cada módulo creado que contendrá, a su vez, dentro de estas se encuentran subcarpetas imprescindibles con los archivos de extensión .PHP y .HTML. De esta forma, podemos controlar los archivos que se encargan tanto del frontend como de la comunicación con la base de datos.

- **Lucidchart:**

Es una herramienta de diagramación y visualización en línea que permite a los usuarios crear diagramas de flujo, mapas mentales, organigramas, diagramas de red, modelos de bases de datos, diagramas UML, entre otros. Entre sus principales características se incluyen colaboración en tiempo real, integraciones con otras herramientas como Google Drive, Microsoft Office.

Esta aplicación fue utilizada para la diagramación del modelo de base de datos (modelo entidad-relación) del sistema en cuestión, esto por su facilidad en el uso, así como su capacidad para representar visualmente información compleja de forma clara y accesible..

Arquitectura de la aplicación:

Flex Mind está diseñado para ser modular, lo que significa que podemos reutilizar componentes de código en diferentes partes de la aplicación. Esto facilita la gestión y el mantenimiento del código, por otra parte está diseñado bajo una arquitectura de cliente-servidor, donde:

- ☐ **Cliente:** es la interfaz que los usuarios utilizan para interactuar con la aplicación y la principal herramienta de los desarrolladores, esta es accesible desde internet y desde cualquier navegador web, lo que permite gestionar archivos, sincronizar con el servidor, personalizar accesos y acciones.
- ☐ **Servidor:** siendo el componente que maneja las solicitudes del cliente y proporciona los datos necesarios. Se encarga de almacenar los archivos, gestionar la

sincronización entre el servidor de desarrollo y el de producción, y asegurar que los cambios realizados por los desarrolladores se reflejan correctamente en la aplicación.

La comunicación entre el cliente y el servidor se realiza a través de protocolos HTTP/HTTPS. Cuando un usuario realiza una acción en el cliente (por ejemplo, guardar un archivo), esta acción se envía al servidor, que procesa la solicitud y devuelve una respuesta al cliente. Esta arquitectura permite que múltiples usuarios trabajen simultáneamente sin interferir entre sí, los archivos sincronizados se marcan en verde, mientras que los no sincronizados se marcan en rojo, facilitando la gestión y el seguimiento de los cambios.

Esta arquitectura de cliente-servidor permite una experiencia de desarrollo eficiente y colaborativa, asegurando que los desarrolladores puedan trabajar de manera efectiva y sin conflictos gracias al uso de este framework.

Características adicionales del sistema:

Algunos detalles técnicos destacables de las aplicaciones desarrolladas con este framework son:

1. **Seguridad:** medidas de seguridad robustas para proteger los datos y archivos de los desarrolladores. Esto incluye autenticación de usuarios, control de acceso basado en roles y cifrado de datos.
2. **Integración Continua:** permitiendo a los desarrolladores automatizar pruebas y despliegues. Esto asegura que los cambios en el código se prueben y desplieguen de manera eficiente y sin interrupciones.
3. **Escalabilidad:** puede manejar un gran número de usuarios y solicitudes sin comprometer el rendimiento. Esto es crucial para aplicaciones que necesitan crecer y adaptarse a una mayor demanda.
4. **Soporte Multiplataforma:** es compatible con múltiples plataformas y dispositivos, lo que permite a los desarrolladores crear aplicaciones que funcionen en diferentes entornos, como Windows, macOS, Linux, y dispositivos móviles.

Descripción de los lenguajes de programación web utilizados en el proyecto:

- **JavaScript:**

JavaScript es un lenguaje de programación interpretado y orientado a prototipos que se utiliza principalmente en el desarrollo web para agregar interactividad y dinamismo a las páginas. Originalmente creado en 1995 para funcionar en el lado del cliente (frontend) dentro de los navegadores, JavaScript permite a los desarrolladores manipular el Document Object Model (DOM) de una página web, lo que facilita la creación de experiencias de usuario interactivas como formularios dinámicos, animaciones, ventanas emergentes, validación de formularios y contenido en tiempo real.

Gracias a su versatilidad, JavaScript es hoy en día uno de los lenguajes más populares y ampliamente utilizados para el desarrollo web y de aplicaciones, permitiendo a los desarrolladores crear aplicaciones completas que van desde sitios estáticos hasta aplicaciones dinámicas y de alto rendimiento.

- **PHP:**

PHP (Hypertext Preprocessor) es un lenguaje de programación de propósito general, especialmente diseñado para el desarrollo web en el lado del servidor. Creado en 1994, PHP es un lenguaje interpretado que permite a los desarrolladores generar contenido dinámico, interactuar con bases de datos y manejar la lógica de servidor para aplicaciones web. Cuando se combina con HTML, PHP es capaz de generar páginas dinámicas, creando y mostrando contenido en función de los datos del servidor o de las solicitudes del usuario.

Componentes del Framework empleados:

- **FlexGrid:** Es el componente encargado de generar las tablas de contenido para visualizar datos. El componente genera las consultas a la base de datos automáticamente, y aplica los formatos a los diferentes tipos de datos.
- **FlexForm:** Es un componente para generar un formulario dinámico en el framework flexmind. Cada formulario tiene diferentes tipos de campos, que incluyen funcionalidades complejas como ser: menú desplegable dinámicos, árbol de contenidos, coordenadas geográficas en google maps, subir archivos, etc.
- **SubGrid:** Dentro del framework flexmind es una subgrilla que muestra información (registros en una tabla secundaria) relacionada a un registro de la tabla principal.
- **SubForm:** Es un formulario secundario o anidado que está relacionado con el formulario principal. Los subforms permiten gestionar y almacenar datos relacionados en una estructura de base de datos que utiliza tablas vinculadas. Esto es útil para manejar relaciones de uno a muchos entre las entidades que se representan en los formularios.
- **Flex Report:** Se utiliza para crear reportes detallados que pueden incluir gráficos, tablas y otros elementos visuales. Es posible extraer datos de la base de datos y presentarlos de manera clara y comprensible. Esto es útil para generar informes de ventas, análisis de rendimiento, y otros tipos de reportes que requieren una presentación visual de los datos
- **UrlValidator:** se trata de un archivo php donde el programador podrá ejecutar validaciones especiales. Estas validaciones se ejecutan antes del insert o update del formulario. Siempre, al validar retornara la estructura XML predefinida donde se visualizarán los errores en caso de haberlos.
- **OnSubmitExecute:** Es un evento o función que se ejecuta cuando se envía un formulario en una aplicación de Flexmind. Este evento permite realizar acciones específicas justo antes o después de que el formulario sea enviado. Se utiliza para validar datos, realizar cálculos, o ejecutar cualquier lógica necesaria antes de que los datos del formulario sean procesados por el servidor.

1.

- **Aplicativos similares:**

- ☐ Plataforma oficial del Gobierno de la Provincia de Formosa:
<https://www.formosa.gob.ar/excursiones>

TODOS UNIDOS

GOBIERNO DE FORMOSA

f

o

t

y

Buscar...

Gobierno

Síntesis Institucional

Trámites y Servicios

Consultas

Turnos

Mi Portal

Menú

Agenda Impositiva

Capacitaciones

Clima y Caminos

Consulados

+ Licitaciones

Medios de Comunicación

Municipios

Transportes

Excursiones

Localidad

Guaycolec

Nombre

Buscar

Guaycolec

Administración Reserva Natural Guaycolec

Dirección: Dirección de Bosques

Teléfono: (54) 0370 - 4436322

- ☐ Plataforma oficial del Ministerio de Turismo:
<http://www.formosahermosa.gob.ar/prestadores-turisticos/>

Formosa

HERMOSA

INICIO

FORMOSA HERMOSA

QUÉ VISITAR

QUÉ HACER

PLANEA TU VIAJE

NOTICIAS TURÍSTICAS

CIUDAD DE FORMOSA

CHE ROGA

Servicios: excursiones.

Teléfono: 3704 557796

Email: cherogaturismo@gmail.com

Web: www.cheroga.com.ar

DOMÍNGUEZ EXCURSIONES

Servicios: guía turístico

Teléfono: 3704 413923

Email: fede_fsa@yahoo.com.ar

Dirección: San Martín 1577

ODISEA FORMOSA "ISLA DE ORO"

Servicios: Turismo aventura.

FERNANDO GAONA

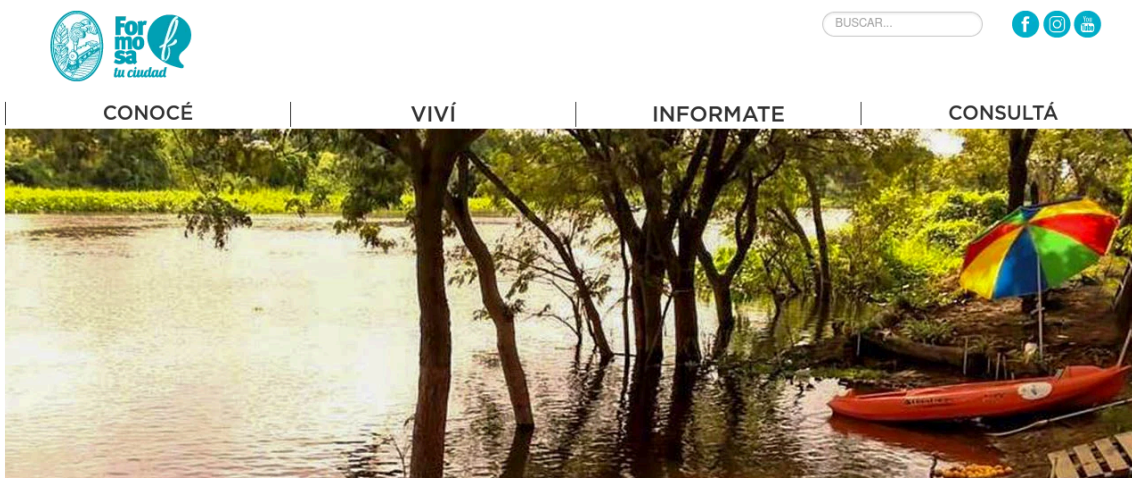
Servicios: Senderismo – Avistaje de Aves – Aventura .

Teléfono: 370 4000077

↑

- ☐ Plataforma oficial de la municipalidad de Formosa:

<https://turismo.formosatuciudad.gob.ar/vivi/circuitos-turisticos/15-turismo-aventura>



- ☐ Pagina de la empresa de turismo Staff: <https://www.staffturismo.com.ar/>



● **Objetivos:**

- ☐ El objetivo del sistema es ofrecer una plataforma de guías turísticas que permita a los usuarios explorar información detallada y organizada sobre distintos puntos de interés en una región, como puntos históricos, paseos culturales, platos típicos, centros de salud, oficinas gubernamentales, y especies autóctonas.

El sistema permitirá la visualización, carga y filtrado de datos por departamento y localidad para cada una de estas categorías, facilitando a los usuarios, tanto a turistas como a habitantes locales, el acceso a una guía personalizada y específica y aprovechar los recursos culturales, históricos y culinarios de cada localidad.

2. **Entidades que intervienen en el sistema:**

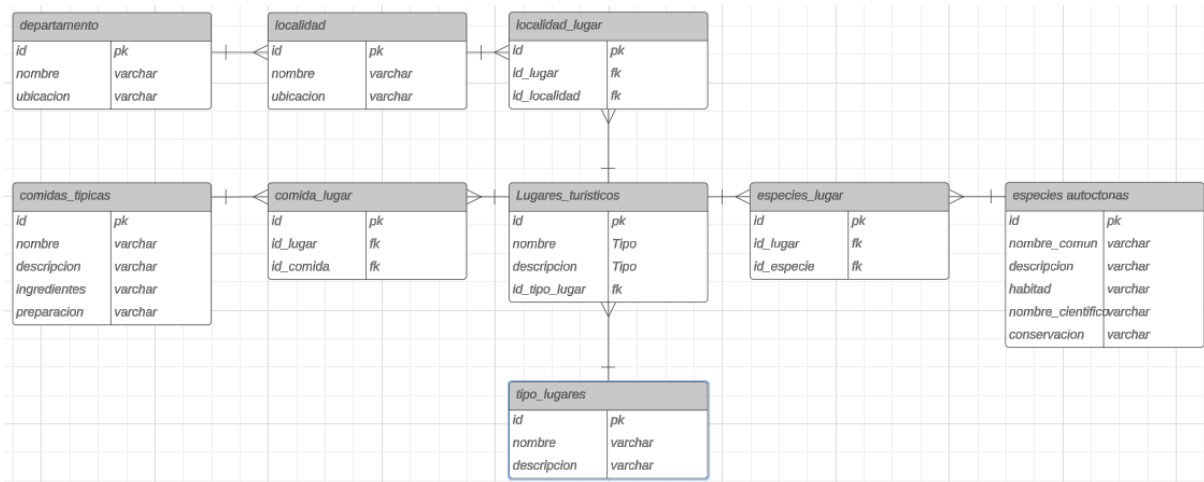
Teniendo en cuenta el objetivo del sistema, las entidades del mismo serían las siguientes:

- Departamentos.
- Localidades.

- Lugares Turísticos
- Tipo de lugares.
- Paseos culturales
- Especies autóctonas..
- Comidas típicas

Diseño y diagramación de la base de datos:

- [lucid.app](#)



3. Plan de acción a seguir durante el proceso de desarrollo:

Opción C: Guia Turistica.

- Definición de requisitos y objetivos:**
Secciones de contenido, como comidas típicas, paseos culturales, etc, se deberán contar con grillas para la visualización de la información, así como también formularios para poder cargar la misma o poder editarla en caso de ser necesario.
- Reconocimiento de las entidades que intervienen:**
El sistema debe contar con una sección de puntos históricos, paseos culturales, comidas típicas, centros de salud, oficinas gubernamentales y especies autóctonas.
- Diseño de la base de datos:** Una vez reconocidas las diferentes entidades, se procederá con el diseño de las relaciones entre estas y posteriormente agregarlas al módulo de base de datos.
- Desarrollo de los módulos correspondientes:** Una vez se cuente con las tablas de la base de datos creadas, se procederá con desarrollar los diferentes módulos correspondientes a las entidades que intervienen en el sistema.
- Integración en el modulo de produccion:**

6. **Pruebas y optimización:**

Una vez que los módulos estén disponibles para ser utilizados, se deberán realizar las pruebas correspondientes para cada uno y así comprobar el correcto funcionamiento de estos, en caso de encontrarse errores, estos deben ser depurados y testeados nuevamente hasta lograr el total funcionamiento.

7. **Implementación:** .

8. **Mantenimiento:** .

4. **Estructura y contenido de la aplicación:**

5. **Diseño y diagramación de la base de datos:**

6. **Explicación de la lógica utilizada para la programación del proyecto:**

7. **Pruebas de funcionamiento:**

8. **Conclusiones sobre el trabajo:**