INSTITUTO TECNOLÓGICO SUPERIOR ZACATECAS NORTE

INGENIERÍA EN SISTEMAS COMPUTACIONALES

DEVOPS

DANIEL ARREDONDO SALCEDO

ESTRUCTURA DEL REPOSITORIO

XITLALIC GUADALUPE FLORES SALCEDO 21010140

MONSERRAT LÓPEZ AGUILAR

BRIAN DIAZ CARRILLO 21010014

RÍO GRANDE, ZACATECAS. AGOSTO 2025

Documentación de la estructura de directorios y nomenclatura de archivos del proyecto MiTec

Este documento describe cómo se organiza el proyecto, qué reglas se siguen para nombrar archivos y carpetas, y cómo se maneja el control de versiones. Su propósito es estandarizar el trabajo y facilitar la colaboración entre nuestros desarrolladores.

```
MiTec/
├── index.html              # Página principal
├── Conocenos/              # Sección "Conócenos"
│   ├── filosofia.html      # Filosofía institucional
│   └── directorio.html     # Directorio académico
├── OfertaEducativa/        # Sección de oferta educativa
│   ├── carreras.html       # Listado de carreras
│   └── modalidades.html    # Modalidades de estudio
├── Servicios/              # Servicios institucionales
│   ├── sce.html            # Sistema de Control Escolar
│   ├── moodle.html         # Plataforma Moodle
│   └── bolsa-trabajo.html  # Bolsa de trabajo
├── FAQ.html                # Preguntas frecuentes
├── assets/                 # Recursos estáticos
│   ├── css/
│   │   └── estilos.css     # Hoja de estilos principal
│   └── img/                # Todas las imágenes del sitio
└── README.md               # Documentación general
```

2. Propósito de cada directorio

- **Conocenos/**: Contiene información institucional como filosofía y directorio.
- **OfertaEducativa/**: Información académica sobre carreras y modalidades.
- **Servicios/**: Accesos y descripción de servicios digitales institucionales.
- **assets/**: Recursos reutilizables (estilos, imágenes, scripts).
- **Archivos raíz**: Páginas principales y FAQ.

3. Estándar de uso del repositorio

3.1 Reglas de nomenclatura de archivos

- Usar **español** para nombres de carpetas y archivos HTML.
- Nombres en **minúsculas**.
- Separar palabras con **guiones medios** (-) si es necesario.
- Estructura clara y descriptiva: nombre-seccion.html.
- Carpeta de imágenes: assets/img/ con nombres descriptivos.

3.2 Reglas de versionado

Se seguirá **Semantic Versioning 2.0.0**:

- **MAJOR**: Cambios que rompen compatibilidad.
- **MINOR**: Nuevas funcionalidades compatibles.
- **PATCH**: Correcciones menores.
  Ejemplo: v1.0.0 para la primera versión estable.

3.3 Convenciones para commits

- Comando utilizado: git commit -am "mensaje descriptivo"
- Formato del mensaje: **Español claro y descriptivo**
- Ejemplos reales utilizados en el proyecto:

    - git commit -am "Realice el archivo servicios.html"
    - git commit -am "Agregue estilos a la sección conocenos"
    - git commit -am "Corregi errores en la página principal"
- **Detalles adicionales**: Los detalles específicos del commit se agregan en la descripción del Pull Request en GitHub.

4. Flujo de trabajo en ramas

- main: Rama principal con código estable y probado.
- Ramas de características: features-nombredelarama (ejemplos reales):
    - features-servicios
    - features-Ofertaeducativa
    - features-conocenos
- Cada colaborador trabaja en su rama feature y hace commits periódicos.
- Al finalizar una sección, se realiza **Pull Request** a main desde GitHub.

5. Proceso de revisión y merge

1. Colaborador completa su sección en la rama features-[nombre]
2. Realiza **Pull Request** desde GitHub
3. En el PR se especifican los detalles técnicos del trabajo realizado
4. **CCB** revisa el código y los detalles del commit
5. Si es aprobado, el CCB realiza el **merge** a main
6. En caso de rechazo, se documentan las razones en el PR