



UNIVERSIDAD ESTATAL AMAZONICA

CARRERA:

Tecnología de la Información

MATERIA:

Estructura de Datos

TEMA:

Identificación De Tipos De Datos

NIVEL:

Tercero

PARALELO:

"D"

ESTUDIANTE:

Carlos Jefferson Cueva Sánchez

DOCENTE:

ING. Santiago Israel Nogales Guerrero

ENLACE GITHUB:

https://github.com/CJCuevas1/P_E_semana-4



INTRODUCCION

La estructura de datos constituye un pilar fundamental en el desarrollo de software moderno. Este concepto se refiere a la organización, gestión y almacenamiento eficiente de información dentro de un sistema computacional. Las estructuras determinan cómo se accede a los datos, se modifican y se relacionan entre sí, impactando directamente el rendimiento y funcionalidad de las aplicaciones.

La presente práctica se enfoca en la implementación de una agenda telefónica utilizando el lenguaje de programación C#. Este ejercicio permite explorar de manera concreta el funcionamiento de colecciones genéricas, específicamente la clase `List<T>`, dentro del paradigma de programación orientada a objetos. La agenda telefónica representa un caso de estudio ideal para comprender la manipulación de datos estructurados en entornos de programación.

El desarrollo de esta solución involucra la creación de una clase `Contacto` que encapsula las propiedades esenciales de cada entrada: nombre, número telefónico y dirección de correo electrónico. Esta abstracción facilita el manejo organizado de la información y demuestra los principios de encapsulamiento propios de la programación orientada a objetos. La implementación se complementa con un menú interactivo que ofrece funcionalidades básicas de gestión, incluyendo visualización completa del directorio y búsqueda específica de registros.



Desarrollo

El proceso de implementación de la agenda telefónica inició con el análisis del problema planteado. Se identificaron los requisitos esenciales: almacenar información de contacto, permitir su visualización completa y facilitar la búsqueda de registros específicos. La solución debía demostrar el uso apropiado de estructuras de datos dentro del paradigma de programación orientada a objetos.

Se seleccionó la clase `List<T>` de C# como estructura principal para almacenar los contactos. Esta decisión se basó en su naturaleza dinámica, su capacidad para crecer según la demanda y su compatibilidad con operaciones frecuentes de adición y recorrido. La alternativa de arrays tradicionales fue descartada por su tamaño fijo y menor flexibilidad para este escenario particular.

La construcción comenzó con la definición de la clase `Contacto`. Esta encapsula tres propiedades fundamentales: `Nombre` (`string`), `Telefono` (`string`) y `Email` (`string`). Se implementó un constructor que acepta estos parámetros, estableciendo valores por defecto para el campo opcional de correo electrónico. Esta abstracción permite tratar cada entrada telefónica como un objeto coherente con identidad y comportamiento definidos.

Posteriormente se diseñó el menú interactivo que ofrece cuatro opciones principales. La primera opción permite visualizar todos los contactos almacenados, mostrando índice, nombre, teléfono y correo electrónico cuando este existe. La segunda función habilita la adición de nuevos registros mediante la captura de datos desde la consola, incluyendo validaciones básicas para campos obligatorios. La tercera opción ejecuta búsquedas por nombre o número telefónico, realizando comparaciones insensibles a mayúsculas y mostrando todos los resultados coincidentes.



La implementación se desarrolló en Visual Studio Code utilizando el SDK de .NET 8.0. El entorno de escritorio proporcionó las herramientas necesarias para compilación, ejecución y depuración incremental. Cada funcionalidad fue probada de manera aislada antes de integrarse al sistema completo, asegurando el comportamiento esperado en diferentes escenarios de uso.

Se incorporó una sección dedicada al uso de inteligencia artificial en el desarrollo. Esta incluye la especificación del agente empleado (ChatGPT) y el porcentaje estimado de código generado mediante esta herramienta. La integración de asistencia por IA permitió acelerar ciertas tareas de implementación mientras se mantenía la comprensión conceptual de cada componente del sistema

El proyecto culmina con un análisis detallado de la estructura List<T>. Este examen considera sus características técnicas, ventajas prácticas para el dominio de agenda telefónica y limitaciones inherentes a su diseño. La reflexión incluye comparaciones con otras estructuras potenciales y justifica la elección final basándose en los requisitos específicos del problema.

Conclusión

La implementación de la agenda telefónica permitió comprender la aplicabilidad concreta de las estructuras de datos en problemas del mundo real. List<T> demostró ser una elección adecuada para este contexto, ofreciendo un equilibrio entre simplicidad de uso y funcionalidad suficiente para las operaciones requeridas.



El ejercicio evidenció las ventajas del paradigma orientado a objetos en la organización de sistemas de software. La encapsulación de datos dentro de la clase Contacto proporcionó claridad conceptual y facilitó la manipulación coherente de la información. Esta aproximación mostró cómo la abstracción mejora la mantenibilidad y extensibilidad del código.

El análisis comparativo reveló que cada estructura de datos presenta compensaciones inherentes entre rendimiento, complejidad y facilidad de implementación. La selección de List<T> sobre alternativas como arrays o Dictionary se justificó por la naturaleza de las operaciones predominantes en una agenda telefónica, donde la adición y visualización completa tienen mayor frecuencia que la búsqueda individual.

La experiencia con herramientas de inteligencia artificial en el desarrollo ilustró tanto sus capacidades como sus limitaciones. Mientras aceleraron aspectos sintácticos y estructurales, la comprensión conceptual y las decisiones de diseño permanecieron como responsabilidad del desarrollador. Esta colaboración resalta la evolución de las prácticas de programación hacia modelos híbridos que combinan creatividad humana con asistencia computacional.

Finalmente, la práctica consolidó la importancia del pensamiento algorítmico en la resolución de problemas computacionales. Más allá de la sintaxis específica de C#, los conceptos de organización de datos, eficiencia y diseño modular demostraron ser transferibles a otros lenguajes y dominios, estableciendo fundamentos para aprendizajes posteriores en el campo de las estructuras de datos.



ANEXOS

AGENDA TELEFÓNICA - UEA

MENÚ PRINCIPAL

1. Ver todos los contactos
2. Agregar nuevo contacto
3. Buscar contacto
4. Eliminar contacto
5. Ver estadísticas
6. Análisis de estructuras
7. Información del proyecto
8. Salir

Seleccione una opción (1-8):

LISTA DE CONTACTOS

[1] Juan Pérez
 Teléfono: 0981234567
 Email: juan.perez@email.com
 Dirección: Av. Amazonas 123

[2] María García
 Teléfono: 0998765432
 Email: maria.garcia@empresa.com
 Dirección: Calle 10 de Agosto

[3] Carlos López
 Teléfono: 0971122334
 Dirección: Barrio Centro

[4] Ana Martínez
 Teléfono: 0964455667
 Email: ana.m@universidad.edu.ec

Presione Enter para continuar...



AGREGAR CONTACTO

Nombre: Peter
Teléfono: 0963892976
Email (opcional): avilespiter35@gmail.com
Dirección (opcional): Av.Las America
 Contacto agregado correctamente.

☞ Presione Enter para continuar...

ELIMINAR CONTACTO

- [1] Juan Pérez
 ✉ Teléfono: 0981234567
 ✉ Email: juan.perez@email.com
 📍 Dirección: Av. Amazonas 123
- [2] María García
 ✉ Teléfono: 0998765432
 ✉ Email: maria.garcia@empresa.com
 📍 Dirección: Calle 10 de Agosto
- [3] Carlos López
 ✉ Teléfono: 0971122334
 📍 Dirección: Barrio Centro
- [4] Ana Martínez
 ✉ Teléfono: 0964455667
 ✉ Email: ana.m@universidad.edu.ec
- [5] Peter
 ✉ Teléfono: 0963892976
 ✉ Email: avilespiter35@gmail.com
 📍 Dirección: Av.Las America

Ingrese el número del contacto a eliminar: 5
 Contacto eliminado exitosamente.

ANÁLISIS DE ESTRUCTURAS DE DATOS

ANÁLISIS DE ESTRUCTURAS DE DATOS UTILIZADAS

1. List<Contacto>

Ventajas:

- Tamaño dinámico.

* - Fácil inserción y eliminación.

Desventajas:

- Búsquedas lineales en grandes volúmenes.

2. Vector (array int[])

Ventajas:

- Acceso directo por índice.

- Uso eficiente de memoria.

Desventajas:

- Tamaño fijo.

- Menor flexibilidad que las listas.

3. Programación Orientada a Objetos

Ventajas:

- Organización del código.

- Reutilización y mantenibilidad.

Desventajas:

- Mayor complejidad inicial.

☞ Presione Enter para continuar...



INFORMACIÓN DEL PROYECTO

Asignatura: Estructura de Datos
Práctica: #01 - Identificación de tipos de datos
Institución: Universidad Estatal Amazónica
Período académico: 2025-2026
Lenguaje: C# (.NET)

▷ Presione Enter para continuar...

ESTADÍSTICAS

- Total de contactos: 4
- Contactos con email: 3
- Contactos con dirección: 3

▷ Presione Enter para continuar...

BUSCAR CONTACTOS

Ingrese criterio de búsqueda: Peter
[1] 🚩 Peter
📞 Teléfono: 0963892976
✉️ Email: avilespt35@gmail.com
📍 Dirección: Av.Las America

▷ Presione Enter para continuar...



BIBLIOGRAFIA

Garay Avendaño, F. (2022). LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS. Introducción a POO. Fase de diseño. Fase de construcción con lenguajes de cuarta generación (C Sharp, Visual Studio, Java Netbeans). Metodología de desarrollo de programas orientados a objetos. Objetos, clases, características principales. Principales sentencias. Comandos controles y sus propiedades. Estructura de control y procedimiento, arrays, polimorfismo, interfaz gráfica. creación y aplicación de sistemas.
<https://repositorio.une.edu.pe/server/api/core/bitstreams/44603984-a2f7-4884-b243-9b9ec127d792/content>

Murillo Valdivia, E. R. (2022). LENGUAJE DE PROGRAMACIÓN II Programación Orientada a Objetos. Objetos. Clases. Herencia. Polimorfismo. Lenguajes. Utilidades para el Desarrollo y Prueba de Programas. Compiladores. Intérpretes. Depuradores. Prueba de Programas en Lenguaje de Programación C ó Java.
<https://repositorio.une.edu.pe/server/api/core/bitstreams/45dfee5f-08ee-4f67-8f69-1686e9d1e287/content>

Salinas Copo, A. M., Vinueza Rodriguez, L. T., Valdiviezo Rodríguez, J. G., Villa López, R. O., Guevara Aulestia, D. L., Urquiza Alvarez, C. E., ... & Espín Mendoza, I. V. (2025). Fundamentos de Programación Orientada A Objetos (POO). <https://hdl.handle.net/10609/149901>