

Universidad San Carlos de Guatemala
Facultad de ingeniería
ingeniería en Ciencias y Sistemas
Ing. Otto Amílcar Rodríguez Costa
Aux. Danny Hugo Bryan Tejaxún Pichiyá



Manual Técnico
Practica 1

Josue Geovany Yahir Perez Avendaño
202300814

Requisitos de Sistema

- Node.js \geq 14.0.0
- Permisos de lectura en directorio de entrada
- Permisos de escritura en directorio src/reports/

Tecnologías Utilizadas

- **Runtime:** Node.js (ES6 Modules)
- **Lenguaje:** JavaScript ES6+
- **Módulos:** ES6 Import/Export
- **I/O:** File System API, Readline
- **Formato de salida:** HTML5 + CSS

Clases y Componentes

1. CallCenterApp (main.js)

Propósito: Punto de entrada de la aplicación

```
export class CallCenterApp {  
  static async main()  
}
```

MenuManager (utils/MenuManager.js)

Propósito: Gestión de interfaz de usuario y navegación

Atributos:

- callCenterManager: CallCenterManager - Instancia del gestor principal
- rl: readline.Interface - Interfaz de entrada de usuario

Métodos principales:

- showMenu(): void
- start(): void
- loadCallDataFromFile(): void
- exportCallHistoryReport(): void
- validateData(): void

CallCenterManager (services/CallCenterManager.js)

Propósito: Lógica de negocio central

Métodos principales:

- loadRecords(records: Array): void
- calculatePerformanceOperators(): Array
- calculateGradePercentage(): Object
- calculateCallsByGrade(): Object

Modelos de Datos

Call (models/Call.js)

- client: Client
- operator: Operator
- stars: number

Client (models/Client.js)

- id: number
- name: string

Operator (models/Operator.js)

- id: number
- name: string
- answeredCalls: number
- incrementAnsweredCalls(): void

FileReader

readFileSync(filepath: string): string

- **Propósito:** Lee contenido completo de archivo
- **Parámetros:**
 - filepath: Ruta absoluta o relativa al archivo
- **Retorna:** Contenido del archivo como string
- **Excepciones:** Error si el archivo no existe

parseCallRecords(content: string): Array<Object>

- **Propósito:** Parsea contenido CSV a objetos JavaScript
- **Formato esperado:** operatorId,operatorName,stars,clientId,clientName
- **Validaciones:**
 - Exactamente 5 campos por línea
 - IDs numéricos
 - Formato de estrellas: x;x;x;0;0 (5 caracteres)

countStars(starsStr: string): number

- **Propósito:** Cuenta estrellas válidas en formato específico
- **Formato:** x;x;x;0;0 donde x=estrella, 0=no estrella
- **Retorna:** Número entero de 0 a 5

ReportGenerator

Template HTML

Cada método generateXXXHtml() genera HTML completo con:

- DOCTYPE HTML5
- CSS
- Tablas con estilos hover
- Codificación UTF-8

Consideraciones de Seguridad

Validación de Entrada

- Sanitización de rutas de archivos
- Validación de formato de datos
- Prevención de inyección de código en HTML

Manejo de Errores

- Try-catch en operaciones de I/O
- Validación de permisos de archivo
- Mensajes de error informativos sin exposición de sistema