

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA



Paradigmas de Programación

Informe N°2

Sistema de gestión fastClinic

Nombre:	Joaquín Ignacio Villagra Pacheco
R.U.N.:	18.847.934-0
Carrera:	Ingeniería Civil Informática
Año estimado de egreso:	2018
E-mail:	joaquin.Villagra@usach.cl
Profesor:	José Allende
Ayudante:	Mauricio Rojas

Santiago, Chile

Viernes, 17 de Octubre de 2014

Índice de contenidos

1	INTRODUCCIÓN	1
2	DESCRIPCIÓN DEL PROBLEMA	2
2.1	Motivación	2
2.2	Especificaciones del proceso de atención	2
2.3	Funcionalidades solicitadas	3
2.4	Dominio del problema	3
3	DESCRIPCIÓN DE LA SOLUCIÓN	4
3.1	Propósito de la solución	4
3.2	Características de la solución	4
3.3	Desafíos de la solución	5
3.4	Funcionalidades implementadas	5
4	ANÁLISIS DE RESULTADOS	6
4.1	Alcance de la solución	6
5	CONCLUSIÓN	6
6	REFERENCIAS	7

1 INTRODUCCIÓN

Un recinto clínico requiere generar un sistema de gestión de sus flujos de información. Se posee una base de datos existente, distribuida en diversos archivos de texto y se solicita a un estudiante de ingeniería informática generar un software, escrito en lenguaje de programación Scheme, bajo el paradigma de programación funcional, que sea capaz de manejar un conjunto definido de funcionalidades. Para ello se le detallan reglas y formatos que deben seguir la compilación, ejecución y entrega de resultados de dicho software.

El presente documento resume el trabajo realizado en el desarrollo del software solicitado, a su vez explica cada funcionalidad implementada y los problemas que surgieron en el desarrollo de las mismas. Este documento se compone de los siguientes capítulos:

- Descripción del problema, sección donde se especifican cada funcionalidad a desarrollar, reglas y formatos exigidos.
- Descripción de solución, sección donde se detalla cada una de las funciones y/o procedimientos desarrollados para cumplir las operatividades exigidas.
- Conclusiones, sección donde se realiza una visión global de lo aprendido, problemas y/o errores presentes en el desarrollo y el cumplimiento de los objetivos planteados.
- Referencias, sección donde se informa cada una de las fuentes de información consultadas.
- Anexo Código Fuente.

2 DESCRIPCIÓN DEL PROBLEMA

2.1 Motivación

Para el recinto clínico *FastClinic* es de imperiosa necesidad manejar su información de una manera sencilla, rápida y centralizada. Su antiguo sistema informático, sin soporte, ya no cubría sus necesidades y por es por esta razón que se ven en la necesidad de solicitar el desarrollo de un nuevo software, teniendo como base de registros archivos de textos heredados del antiguo sistema.

2.2 Especificaciones del proceso de atención

Para el desarrollo del nuevo sistema, la directiva de *FastClinic* nos ha entregado cierta información relevante, referente al proceso de atención de los pacientes y las relaciones existentes entre los distintos datos que se registran.

1. Un paciente es atendido por uno o más médicos y un médico puede atender múltiples pacientes.
2. Un paciente puede llegar con uno o varios motivos de consultas.
3. Un paciente puede tener uno o varios diagnósticos.
4. Los diagnósticos son efectuados por un médico responsable en una fecha (dd/mm/aaaa) particular.
5. Los pacientes pueden recibir distintos tratamientos recomendados por un doctor. Los tratamientos están normalizados para cada diagnóstico.
6. Los diagnósticos pueden ser tratados con uno o más tratamientos. Los tratamientos pueden ser comunes para uno o más diagnósticos.

7. Los pacientes reciben el alta de un diagnóstico en una fecha (dd/mm/aaaa) particular y bajo la responsabilidad de un médico (que puede ser diferente del médico que hizo el diagnóstico original).

2.3 Funcionalidades solicitadas

Dentro del enunciado del proyecto a desarrollar, se detallan diversas funcionalidades que pueden ser implementadas. No se establece que deben implementarse todas, quedando a criterio del tiempo del estudiante decidir cual programar. En el siguiente capítulo, se especifican las funcionalidades implementadas como parte de la solución. En el caso que se quiera conocer el abanico de funciones que podrían haber sido implementadas, favor dirigirse al enunciado del proyecto.

2.4 Dominio del problema

Por parte del estudiante de ingeniería, existe un control medio del problema ya que gracias a lo aprendido en cursos anteriores puede aplicar la lógica de resolución de problemas, identificando las similitudes fuertes o debiles entre las diversas funcionalidades solicitadas para, de esta forma, aprovechar de mejor manera el tiempo invertido. Por otra parte, el manejo del lenguaje scheme es una falencia, ya que estos contenidos, aunque han sido entendidos en las clases de la asignatura, no se han interiorizado en gran medida. Notemos que el cambio de paradigma es algo complicado en primera instancia y esto, genera retrasos de implementación en el proyecto.

3 DESCRIPCIÓN DE LA SOLUCIÓN

3.1 Propósito de la solución

El objetivo principal es generar un software de gestión que cumpla los estándares especificados por la gente de FastClinic y que sea capaz de entregar satisfacción al usuario, optimizando el tiempo invertido en los procesos productivos de dicha entidad.

3.2 Características de la solución

El software generado estará codificado en lenguaje Scheme, bajo las reglas de la programación funcional. Se debe destacar que dicho paradigma de programación utiliza el tipo de dato abstracto Lista y operaciones recursivas como sus piedras angulares de implementación y procesamiento.

Un tipo de dato abstracto, es una implementación de código que permite emular el comportamiento que tiene un "entidad" o "ser" a nivel real. En otras palabras, permite generar una versión programada de los atributos y las acciones que desempeña un determinado individuo. Para este caso, se han implementado los siguientes tipos de datos abstractos:

1. TDA Doctor
2. TDA Paciente
3. TDA Tratamiento
4. TDA Diagnostico
5. TDA DiagnosticoPaciente
6. TDA TratamientoDiagnostico

7. TDA TratamientosDiagnosticosPaciente

3.3 Desafios de la solución

El principal problema o desafio, para el estudiante de ingeniería informática, es poder interiorizar de buena forma los conceptos y la lógica de la programación funcional demostrando, dicho aprendizaje de contenidos, de forma practica a través de la creación del software solicitado.

3.4 Funcionalidades implementadas

A continuación se detallan cada una de las funcionalidades trabajadas. Se debe destacar que el total de funcionalidades implementadas son 9, distribuyendose de la siguiente forma:

- 5 Funcionalidades de nivel bajo.
 1. obtenerNombrePaciente
 2. especialidad
 3. tratamientoRiesgoso
 4. cantidadMedicosTratamiento
 5. cantidadPacientesMedico
- 2 Funcionalidades de nivel medio.
 1. tratamientoMasUsadoPorDiagnostico
 2. medicoMasAltas
- 2 Funcionalidades de nivel alto.
 1. listarMedicosTratantesPaciente
 2. diagnosticoPacienteMedico

- Nota: Además de estas funcionalidades, se implementaron:
 - Selectores
 - Constructores
 - Modificadores
 - Verificadores de pertenencias

Para ver en detalle las implementaciones desarrolladas, favor dirigirse al Anexo de código fuente, adjunto al final del presente documento.

4 ANÁLISIS DE RESULTADOS

En el desarrollo, solo se desarrollaron las funcionalidades que a juicio del estudiante fueran abordables con respecto al tiempo reducido con el que disponía. Es por esta razón que los resultados, teniendo en cuenta única y exclusivamente las funcionales trabajadas, no cumplen con lo esperado. Sin embargo, cada funcionalidad desarrollada, logra dar solución a lo que se solicitó por parte de FastClinic.

4.1 Alcance de la solución

En concreto, se alcanza a cumplir y sobrepasar en baja intensidad el mínimo exigido. Este alcance no satisface al estudiante, sin embargo, es realmente lo que se implementa.

5 CONCLUSIÓN

Durante el desarrollo del software, se presentaron diversas complicaciones. Por una parte, los errores de redacción del enunciado del proyecto propiciaron confusiones del estudiante en cuestión y generaron retardos en la implementación de las funcionalidades solicitadas. Por otro lado, el cambio de paradigma es difícil de comprender, recordemos que el efecto paradigma se hace presente cada vez que intentamos mirar las cosas desde otra perspectiva y nos impide salir de las reglas actuales, en síntesis, nos cuesta reconocer otras formas de hacer las cosas.

En cuanto al lenguaje de programación Scheme, pese a ser un lenguaje antiguo no se queda atrás en poder de procesamiento. Esto puede verse muy bien en la aplicación que tuvo este lenguaje en manejo de grandes tamaños de datos. Su papel en la actualidad es un tanto desconocido, pero no hay duda de que es un lenguaje poderoso aunque de difícil lectura y comprensión.

Finalmente, se puede visualizar bajo la propia experiencia las diferencias entre los distintos paradigmas de programación, teniendo ahora como comparar objetivamente, en un ambiente real las ventajas y desventajas que nos otorga cada paradigma en cuestión.

6 REFERENCIAS

- Andrés Marzal, Isabel Garcia. (Año no informado). Introducción a la programación con Scheme. Departamento de lenguajes y sistemas informáticos: Universitat Jaume.
- Eduardo Navas (2010). Programando con Racket 5. El Salvador: Universidad Centroamericana "Jose Simeón Cañas".