ESCUELA DE INGENIERÍA INFORMÁTICA

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

Programación Avanzada

INF2236-2

Profesores:
Claudio Cubillos
Leonardo González

Integrantes:
Agustín Guzmán
Nicolás Leiva
Felipe Márquez

AVANCE

SIA1.1 Análisis de los datos y principales funcionalidades

El proyecto se centra en la creación de un calendario para el fácil manejo de los quehaceres diarios.

Esto se logra haciendo concretos dos conceptos abstractos: El evento y el calendario. El evento es el hecho puntual que le interesa al usuario recordar (inspirado en el estándar RFC-5545), mientras que el calendario colecciona y gestiona dichos eventos.

Lo anterior se logra con las siguientes funcionalidades:

1. add(): Añade un evento al calendario (respetando un criterio de orden).
2. remove(): Remueve un elemento del calendario.
3. searchByTitle(): Busca un evento del calendario por su título.
4. searchByDate(): Busca un evento del calendario por su fecha.

Además, los eventos contiene un mapa de sus participantes el cuál es modificado desde una interfaz.

Por otro lado, los participantes se gestionan por medio de una colección diferente, para mantener constancia de las personas registradas en el sistema. Estos se gestionan por un mapa que considera las siguientes funcionalidades:

1. add(): Añade una persona a la colección.
2. get(): Obtiene una persona de la colección.
3. remove(): Remueve una persona de la colección.
4. contains(): Avisa si la persona ya está registrada en la colección (vía RUT).
5. isEmpty(): Avisa si la colección está vacía.

SIA1.2 Diseño Conceptual de Clases de Dominio y su Código en Java

Clases Pricipales del Proyecto

Class Main

Main solo hace 2 cosas:

1. Inicializar variables (calendario y personas).
2. Arrancar la interfaz de terminal.

Módulo: Calendario

Class Calendar

Calendar es capaz de gestionar los eventos del programa mediante operaciones CRUD.

Campos:

1. events (List<Event>): Recopilación de todos los eventos.

Operaciones:

1. getAllEvents() → List<Event>: Obtiene todos los eventos del calendario.
2. cmpEvents(Event, Event) → int: Criterio de ordenamiento.
3. add(Event) → boolean: Añade un evento al calendario.
4. remove(Event) → Event: Remueve un evento del calendario.
5. searchByTitle(String) → List<Event>: Retorna una lista de coincidencias.
6. searchByDate(LocalDate) → List<Event>: Retorna una lista de coincidencias.

Class Event

Event contiene todos los campos que definen un evento.

Campos:

1. title (String): Título.
2. date (LocalDate): Fecha.
3. startTime (LocalTime): Hora de inicio.
4. endTime (LocalTime): Hora de termino.
5. isAllDay (boolean): Indicador de si el evento dura todo el día.
6. location (String): Lugar del evento.
7. description (String): Detalles del evento.
8. participants (Recurrence): Las repeticiones del evento.
9. config (CalendarConfig): Contiene las configuraciones del calendario.
10. END_OF_DAY (LocalTime): Definición propia de la hora final del día.

Operaciones:

1. Getters para todos los campos.
2. Setters para todos los campos.
3. Formateadores para poder imprimir fecha y hora.
4. addParticipant(Person person) → boolean: Añade un participante al registro.
5. removeParticipant(String rut) → void: Remueve un participante del registro.
6. getParticipants() → Map<String, Person>: Obtiene a todos los participantes.

Módulo: People

Class People

People gestiona las personas registradas en el sistema mediante operaciones CRUD.

Campos:

2. people (Map<String, Person>): Recopilación de todas las personas.

Operaciones:

1. add(Person) → void: Añade una persona al mapa.
2. get(String) → Person: Obtiene una persona del mapa.
3. remove(String) → void: Remueve una persona del mapa.
4. contains(String) → boolean: Avisa si la persona ya está regsitrada.
5. isEmpty() → boolean: Avisa si la colección está vacía.

Class Person

Person contiene todos los campos que definen a una persona.

Campos:

1. rut (String): Identificador de la persona.
2. name (String): Nombre.
3. email (String): Correo Electrónico.
4. phone (int): Número de teléfono.

Operaciones:

1. Getters para todos los campos.
2. Setters para todos los campos.

Clases Auxiliares Del Proyecto

Módulo: UI

Class UI

UI es tan solo una colección de todas las interfaces que el programa contiene: CalendarUI y PeopleUI. Además, acá se inicializan un Scanner, CalendarUI, PeopleUI y un booleano que representa si la interfaz está corriendo o está detenida.

Es por medio de esta interfaz que se selecciona con qué se pretende trabajar, si con el calendario o con el registro de las personas. Dichas Operaciones son las siguientes:

1. Seleccionar calendario.
2. Seleccionar personas.
3. Salir de la interfaz.

Class CalendarUI

CalendarUI accede a la administración del calendario con ayuda de las operaciones previamente descritas. Además, acá se inicializan un Calendar, People, Scanner y se cargan configuraciones propias del calendario.

Es por medio de esta interfaz, que se pueden hacer las siguientes operaciones:

1. Listar todos los eventos.
2. Buscar un evento (por título o fecha).
3. Añadir un evento.
4. Modificar un evento.
5. Remover un evento.
6. Salir de la interfaz.

Class PeopleUI

PeopleUI accede a la administración de las personas registradas con ayuda de las operaciones previamente descritas. Además, acá se inicializan un Scanner y People.

Es por medio de esta interfaz, que se pueden hacer las siguientes operaciones:

1. Listar todas las personas.
2. Buscar una persona por medio del RUT.
3. Añadir una persona.
4. Modificar una persona.
5. Remover una persona.
6. Salir de la interfaz.

Módulo: Config

Class Config

Config puede cargar y retornar las configuraciones del sistema desde un archivo TOML. Para este propósito, se incicializa una clase CalendarConfig (que guarda las configuraciones específicas del calendario) y la ruta del archivo de configuración.

Lo anterior se logra con las siguientes operaciones:

1. load() → Config: Carga la configuración desde el archivo *settings.toml*.
2. GetCalendar() → CalendarConfig: Retorna las configuraciones del calendario.

Class CalendarConfig

CalendarConfig gestiona las configuraciones del calendario.

Campos:

1. dateFormat (String): Formato de fecha.
2. timeFormat (String): Formato de hora.

Operaciones:

1. Getters para todos los campos.
2. Setters para todos los campos

SIA1.3 Todos los Atributos de Todas las Clases Deben Ser Privados Y Poseer Sus Respetivos Setters Y Getters.

Esto se logra en:

- Ruta: src/ui/modules/PeopleUI.java
- Lineas: 9-10, no corresponden setters ni getters

- Ruta: src/ui/modules/CalendarUI.java
- Lineas: 19-25, no corresponden setters ni getters

- Ruta: src/modules/calendar/Calendar.java
- Lineas: 9, 16-8, 38-59

- Ruta: src/modules/calendar/Event.java
- Lineas: 14-24, 37-156, 178-197

- Ruta: src/modules/people/People.java
- Lineas: 7, 13-28

- Ruta: src/modules/people/Person.java
- Lineas: 3-7, 18-73

- Ruta: src/config/Config.java
- Lineas: 10-12, 14-36

- Ruta: src/config/modules/CalendarConfig.java
- Lineas: 4-5, 8-38

SIA1.4 Deben Incluir Datos Iniciales Dentro Del Código

No implementado.

SIA1.5 Diseño Conceptual y Codificación de 2 Colecciones de Objetos. En la Segunda Colección Anidada

Esto se logra combinando las siguientes clases: Calendar, Event, Person. Se procede a explicitar este flujo:

1. Calendar contiene un campo List<Event> para gestionar los eventos con un ArrayList.

   Ruta: src/modules/calendar/Calendar.java

   Lineas: 9, 12-18

2. Event contiene un campo Map<String, Person> para gestionar sus participantes con un HashMap.

   Ruta: src/modules/calendar/Event.java

   Lineas: 21, 196

3. Person tan solo contiene campos String y un campo int.

   Ruta: src/modules/people/Person.java

   Lineas: 4-7

SIA1.6 Diseño Conceptual y Codificación de 2 Clases Que Usen Sobrecarga de Métodos

Esto se logra en:

- Ruta: src/ui/modules/PeopleUI.java
- Motivo: Poder modificar una persona conociendola de antemano o no.
- Lineas: 105-120

- Ruta: src/modules/Peolpe/Person.java
- Motivo: Poder asignar valores por defecto a los campos.
- Lineas: 35-73

- Ruta: src/modules/Calendar/Event.java
- Motivo: Poder asignar valores por defecto a los campos.
- Lineas: 120-156

- Ruta: src/config/modules/CalendarConfig.java
- Motivo: Poder asignar valores por defecto a los campos.
- Lineas: 19-38

SIA1.7 Diseño Conceptual y Codificación de 1 Clase Mapa

Esto se logra en:

- Ruta: src/modules/people/People.java
- Lineas: 7

- Ruta: src/modules/calendar/Event.java
- Lineas: 21

SIA1.8 Incluir Menú Donde Se Incluyan funcionalidades de interacción (Inserción y Listar)

Esto se logra en:

- Contexto: Agregar participantes de un evento
- Ruta: src/modules/Calendar/Event.java
- Lineas: 444-457

- Contexto: Mostrar participantes de un evento
- Ruta: src/modules/Calendar/Event.java
- Lineas: 84, 117, 131, 257, 491

SIA1.9 Todas las Funcionalidades Deben ser Implementadas Mediante Consola

Esto se logra en:

- Contexto: UI es la colección de interfaces del programa
- Ruta: src/ui/UI.java

- Contexto: CalendarUI es la interfaz para el manejo del calendario
- Ruta: src/ui/modules/CalendarUI.java

- Contexto: PeopleUI es la interfaz para para el manejo de personas
- Ruta: src/ui/modules/PeopleUI.java

SIA1.10 Utilización de GitHub

Esto se logra en: https://github.com/Schiz0idCat/calendar

Coevaluación Avance

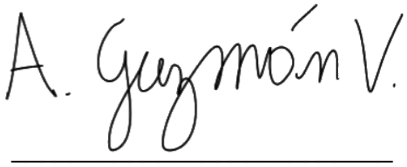
Integrante	Tarea	Nota Final
Agustín Guzmán	Arquitectura del proyecto, configuraciones, class People y Person	7.000
Nicolás Leiva	class Event y class CalendarUI	5.50
Felipe Márquez	Class Calendar y class CalendarUI	6.650

Coevaluación Agustín Guzmán			
Integrante	% Realización	Nota (1-7)	Ponderación
Nicolás Leiva	100%	4	4
Felipe Márquez	100%	7	7

Coevaluación Nicolás Leiva			
Integrante	% Realización	Nota (1-7)	Ponderación
Agustín Guzmán	100%	7	7
Felipe Márquez	90%	7	6.3

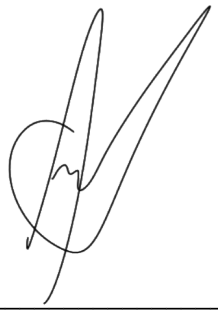
Coevaluación Felipe Márquez			
Integrante	% Realización	Nota (1-7)	Ponderación
Nicolás Leiva	100%	7	7
Agustín Guzmán	100%	7	7

Explicación
Tarea:
Qué tarea hicimos en el trabajo
Nota Final:
Promedio de las coevaluaciones
% Realización:
% de realización de la tarea asignada
Nota Coevaluada:
Nota de calidad de la tarea asignada
Ponderación:
Nota final de la coevaluación



Agustín Guzmán

Nicolás Leiva



Felipe Márquez

FINAL

SIA2.1 Diseño de Diagrama UML

No implementado

SIA2.2 Persistencia de Datos

No implementado

SIA2.3 Implementa Interfaz Gráfica de Ventana

No Implementado

SIA2.4 Interfaz Modificar y Eliminar Elementos. Para la Segunda Colección de Objetos

Esto se implementa en:

- Contexto: La modificación de la persona se hace desde PeopleUI, no desde CalendarUI.
- Ruta: src/ui/modules/PeopleUI.java
- Lineas: 122-182

- Contexto: La eliminación de la persona se puede desde PeopleUI (para eliminarla del registro) y desde CalendarUI (para eliminarlo de un evento).
- Ruta: src/ui/modules/PeopleUI.java
- Lineas: 184-205
- Ruta: src/ui/modules/CalendarUI.java
- Lineas: 459-475

SIA2.5 Incluir Al Menos 1 Funcionalidad Ajena a la Lógica de Negocio

No implementado

SIA2.6 Buena Modularización y Prácticas de Documentación Interna y Legibilidad

Está todo minusiosamente modularizado siguiendo buenas prácticas. Además la Documentación y legibilidad del código tienen el mismo cuidado

SIA2.7 Diseño y Codificación de Dos Clases Con Sobreescritura de Métodos

No implementado

SIA2.8 Implementar Manejo de Excepciones Mediante Try-catch

Esto se logra en:

- ruta: src/ui/modules/CalendarUI.java
- lineas: 44-50, 98-104, 174-180, 224-230, 294-300, 307-312, 321-327, 334-340, 347-353, 360-370, 420-426, 499-505, 529-534, 547-551

- ruta: src/ui/modules/PeopleUI.java
- lineas: 27-33, 95-102, 134-140, 165-171

- ruta: src/ui/UI.java
- lineas: 27-33

- ruta: src/config/Config.java
- lineas: 21-35

SIA2.9 2 Clases que Extiendan de una Excepción y Que Se Usen en el Programa

No implementado.

SIA2.10 Generar Reportes de las Colecciones de Objetos

No implementado.

SIA2.11 Continuidad en el Uso de GitHub

Esto se logra en: https://github.com/Schiz0idCat/calendar

SIA2.12 Implementar Funcionalidades de Agregación, Eliminación y Modificación en Elementos de Primer Nivel

Esto se logra en:

- Contexto: Agregación de eventos
- Ruta: src/modules/calendar/Calendar.java
- Lineas: 38-49

- Contexto: Agregación de personas
- Ruta: src/modules/calendar/People.java
- Lineas: 17-24

- Contexto: Eliminación de eventos
- Ruta: src/modules/calendar/Calendar.java
- Lineas: 51-59

- Contexto: Eliminación de personas
- Ruta: src/modules/calendar/People.java
- Lineas: 26-28

- Contexto: Modificación de eventos
- Ruta: src/modules/calendar/CalendarUI.java
- Lineas: 268-402

- Contexto: Modificación de personas
- Ruta: src/modules/calendar/PeopleUI.java
- Lineas: 122-182

SIA2.13 Implementar la Funcionalidad de Buscar en al Menos un Nivel

Esto se logra en:

- Ruta: src/modules/calendar/CalendarUI.java
- Lineas: 61-88

- Ruta: src/modules/people/People.java
- Lineas: 34-36

Coevaluación Avance

No implementado