

## Trabajo Práctico Final

### Introducción

El trabajo práctico final consta de la realización de una aplicación administradora de calendarios la cual pueda interactuar con diferentes servicios de internet. La aplicación deberá poder visualizar, crear y modificar eventos de un calendario para un usuario particular, persistiendo información de manera interna para su acceso fuera de línea.

### Requerimientos funcionales

#### General

La interfaz del usuario debe ser consistente y no deberá tener errores de interacción ni de visualización de información. El programa deberá ser fácil de usar e intuitivo.

#### Administración de cuentas

La aplicación deberá ser capaz de dar de alta, modificar, eliminar y visualizar roles o usuarios los cuales cada uno podrá tener cuentas Calendario de diferentes servicios. Es decir un usuario puede tener dos calendarios de Google y uno de otro proveedor. Esta información deberá ser persistida internamente por el programa. Los datos a persistir son:

- Nombre: Nombre de la cuenta. Es un dato unívoco, es decir se debe validar que solamente haya uno con el mismo nombre.
- Identificador: Dato identificador del servicio asociado a la cuenta. Esto dependiente del servicio puede ser una cuenta de correo o una clave alfanumérica.
- Servicio: Nombre del servicio que provee las funciones de calendario, tal como Google. El usuario podrá seleccionarlos mediante una lista desplegable o similar.

#### Visualización y obtención de eventos

La aplicación deberá tener una pantalla en donde el usuario podrá visualizar y disparar la recepción de eventos o citas de las cuentas que están configuradas. Se debe presentar al usuario una interfaz visual la cual exponga de forma tabular los eventos del mismo para todos los días del corriente mes.

La aplicación deberá poder de forma manual o automática obtener la información de eventos desde los servicios del usuario y persistirlos internamente para su lectura cuando no se disponga de conectividad.

La falla en el acceso a los datos de una cuenta no debe invalidar el proceso general de obtención de información y debe mostrarse en pantalla que el servicio ha tenido un error de acceso.

El usuario podrá cuando quiera volver a obtener los eventos con el fin de actualizar la información mostrada. Toda información de error debe ser mostrada al usuario de una

forma comprensible.

### **Creación, modificación y eliminación de eventos**

El usuario dentro de la pantalla de visualización de eventos, o de cualquier otra forma, deberá poder crear, modificar y eliminar eventos asociados a sus cuentas. Estas modificaciones deberán sincronizarse con el servicio apropiado de la cuenta y de forma interna de la aplicación.

### **Requerimientos no funcionales**

- El programa deberá tener una interfaz gráfica desarrollada en Swing, integrando los conocimientos y técnicas adquiridos durante la cátedra. La incorporación de conocimientos no adquiridos durante la cátedra serán también bienvenidos.
- Para el acceso a fuentes de calendario utilizaremos los servicios de Google, la librería y documentación puede obtenerse desde la siguiente dirección.

#### **Google Calendar API**

<https://developers.google.com/google-apps/calendar/>

- Cada tipo de servicio de calendario posee sus propias características y configuraciones que no es necesaria que posea una interfaz gráfica, puede ser mediante archivos de configuración o incrustada en el código. Si bien solamente es necesario un servicio la aplicación deberá estar preparada para poder incorporar de manera sencilla otros servicios mediante codificación. Por lo que las mencionadas configuraciones deben estar abstraídas.
- El programa deberá persistir las configuraciones en una base de datos relacional. Se espera que en un futuro puedan configurarse persistencia en distintos gestores de Bases de Datos u otras formas de persistencia por lo que el software debe estar preparado para ello.
- El programa deberá estar correctamente comentado y documentado en formato JavaDoc.

### **Entregables**

Los entregables del trabajo final consisten en:

- Código ejecutable.
- Código fuente.
- Documentación en formato JavaDoc.
- Diagrama de clases y paquetes en formato digital de alguna herramienta UML conocida.
- Guía de Instalación y Uso del programa.

Importante: Se debe entregar los mencionados elementos en un CD.

## **Evaluación**

Los puntos a evaluar del trabajo práctico final son el uso de buenas técnicas de desarrollo y documentación, legibilidad del código, utilización correcta del lenguaje y finalmente la resolución empleada para satisfacer los requerimientos.

Se deberá realizar una defensa grupal e individual del trabajo en el momento de la entrega, en donde los alumnos deberán explicar la solución empleada y responder a preguntas de forma individual del docente. Si bien las tareas de codificación pueden dividirse dentro del grupo, cada integrante debe conocer y comprender de forma completa el desarrollo ya que cualquier integrante tiene que poder responder cualquier pregunta sobre la aplicación.