**Trabajo Practico Final**

CALENDARIO

**Materia:** AYDOO

**Año:** 2017

**Profesor:** Nicolas Paez

**Integrantes:**

* Hernán Cortes
* Martin Capocasale

**Lenguaje de Programación**: Ruby versión 2.2.0

**Decisiones de Diseño:**

Se comenzó programando lo que en un primer momento observamos que tenía mayor complejidad, una clase Archivo que básicamente creara archivos, leyera su contenido y lo mostrara.

Los calendarios serían archivos que contendrían los eventos como texto.

Luego se pensó que para controlar la existencia de los calendarios debía existir un archivo llamado listado\_de\_calendarios que contuviera siempre la lista de los calendarios existentes. No se podía verificar la existencia de los calendarios solo por la existencia de un archivo dentro de la carpeta ya que la carpeta podría contener múltiples archivos de otra índole y no calendarios.

Luego se comenzó el tratamiento de los post, get y put a través de Sinatra. Se crearon todas las funciones que manejaban esos pedidos y se resolvió todo en la propia App.rb, llamando eventualmente a la clase Archivo para generar o mostrar el contenido de cada Calendario.

Luego fue necesario crear la clase Evento, no en particular para guardar los mismos o devolverlos, ya que cada uno de ellos es un json que se guarda en una línea del archivo, por lo cual puede recorrerse el archivo y tomarse uno por uno y analizar su contenido, sino que la necesidad principal de crear Evento fue al solicitar modificar los mismos. Allí no era solo necesario cambiar el contenido, sino también analizar y comparar el evento viejo y el evento nuevo (con el nuevo contenido a guardar) para poder validar que los cambios fueran correctos.

Nos faltó dejar operativa la recurrencia. Preparamos todo para poder implementarla, con la clase Gestor de Recurrencia, pero a último momento no funcionó el cálculo de la misma y decidimos quitar directamente las clases, dejando comentado el comportamiento que no llegamos a implementar.

También vemos que hubiera sido muy bueno implementar patrones correspondientes dentro de la clase Calendario para poder estilizar el diseño, unificando métodos con comportamientos similares.

Vemos la clase Calendario sobrecargada, pero funciona todo lo pedido respecto a crear, modificar, buscar por id, etc. Con más tiempo hubiera sido bueno refactorizar más cosas para obtener un mejor diseño y respetar más los principios SOLID.