

Pontificia Universidad Católica de Chile

Diseño Detallado de Software IIC2113

Noviembre 2014

---

# ENTREGA FINAL: SUBGRUPOS

## COMPONENTE FRONTEND

Integrantes

Tomas Dussailant

Guillermo Morales

Macarena Peralta

## CAMBIOS EN EL DISEÑO

Durante el último periodo de desarrollo no se realizó ningún cambio importante en el diseño del componente. Como funcionalidades se agregan nuevas fuentes pero el manejo de éstas en el componente frontend es muy parecido al que se tenía anteriormente, por lo que no fue necesario crear nuevas clases.

Se trabajó principalmente en desplegar correctamente los resultados, navegar el clouddtag y hacer las llamadas necesarias a la aplicación backend. Para la navegación se incorporaron nuevas vistas parciales y archivos javascript con llamadas Ajax. Todo esto estaba contemplado en el diseño original y no se necesitaron nuevas clases o subcomponentes.

Las razones porque el diseño no cambió para esta entrega se pueden resumir en tres:

1. **El estado de la entrega.** Para esta última iteración del desarrollo se esperaba que no existieran cambios mayores, sino trabajar y refinar las funcionalidades creadas anteriormente.
2. **Trabajo en conjunto con el componente backend.** A diferencia de la entrega anterior se trabajó lado a lado con el grupo encargado del controlador del backend para que cualquier modificación que se necesitara se hiciera de la forma más óptima para toda la aplicación y que cada responsable modificara su propio código. Esto ayudó también a que al juntar los componentes se encontraran menos errores.
3. **El diseño original.** El diseño original contemplaba la opción de extensibilidad, es decir, incorporar nuevas fuentes. Desde un comienzo se dividió la aplicación en distintos subcomponentes con sus propias responsabilidades. Así, por ejemplo, mashup\_controller se encarga de todo el despliegue del mashup sin importar el número de fuentes externas. User\_controller, por otro lado se encarga del manejo de sesión y datos de usuario. Tener las responsabilidades separadas permitió que los nuevos cambios se desplegaran sin problemas mayores por parte del componente frontend.

En conclusión, la aplicación entera funciona bajo los principios de ocultamiento, alta cohesión y bajo acoplamiento. Cada componente no conoce cómo funciona el otro, no es necesario saber cómo funciona el parser para poder ser llamado desde el controlador, y así mismo el backend no conoce como despliega los resultados el frontend. Lo importante es la comunicación a través de interfaces claras, que en el caso del frontend es una API. La cohesión se ve reflejada en las responsabilidades únicas de cada componente y subcomponente, y el acoplamiento se intentó mantener bajo a través de las llamadas vía API y Ajax.