# Presentación técnica

## Objetivos

* Integración de información proveniente de diversas fuentes
* Control de acceso
* Acceso remoto a datos almacenados en la base de datos a través de consultas
* Análisis de datos sobre conductores
* Automatización de procesos de actualización y recuperación de la base de datos
* Administración de cuentas de usuarios (clientes, administradores y operadores)
* [Usabilidad]

## Objetivos de diseño

* Minimización de acoplamiento
* Reutilización de componentes
* Facilidad de extensión de funcionalidad
* …

## Frameworks y tecnologías

(logo de cada cosa)

* Java
* Wicket
* JXL
* Tomcat
* Maven
* Hibernate
* MySQL
* Eclipse
* Log4j
* SVN

## Componentes

## Despliegue

## Arquitectura

[diagrama]

### Ventajas

* Multiplataforma
* Actualización dinámica de la aplicación (el sitio web al que accede el cliente siempre está actualizado)
* Independencia del sistema operativo
* Independencia de la base de datos
* No se requiere instalación
* No es necesario aplicar seguridad en los equipos de clientes

## Modelo de datos

[diagrama]

Aca se puede explicar algo del formato de los archivos de actualización.

## Interfaces con otros sistemas

[Diagrama]

* Descripción de protocolos. Se puede explicar uno de los del doc como ejemplo.

## Seguridad

* URL encriptado
* Control de acceso a través de sesiones de usuario.
* Definición de perfiles que determinan qué provilegios tiene cada tipo de usario.
* Validación de campos
* [Seguridad en la base de datos, MD5, DES, TRIPLE DES algo así??? Y GRANTS]
* [ Si logro agregarlo] Completely Automated Public Turing test to tell Computers and Humans Apart (Captcha)

La implementación de captchas, URLs encriptadas y la validación de campos permite evitar:

* SQL injection. [No es posible especificar como parámetros consultas SQL en el URL o en campos.]
* Cross Site Scripting (XSS). [No es posible ejecutar código a través de campos o URLs.]
* Spam en el formulario de contratación de servicios.

## Mejoras