

Taller de Sistemas de Información 2

Trabajo Obligatorio - Edición 2015

1. Introducción

Potenciado por los avances en las tecnologías de la información se está viviendo una segunda ola de innovaciones en el segmento de sitios generadores de sistemas. Hoy deben ser instalables en una amplia gama de dispositivos, y resulta interesante que todos interactúen con el mismo núcleo. En los últimos meses han surgido una serie de nuevos competidores en lo que era un mercado hasta hace poco dominado por unos pocos.

Se debe buscar que los sistemas tengan un diseño que tome sentido en cualquier de los escenarios de uso que se dan en el mundo real. Desde un teléfono, pasando por una tableta, hasta con una PC clásica. Sacando provecho de lo "transportable" de una tableta pero a la vez que la aplicación tenga sentido en un PC de escritorio, que la aplicación sea portable y tenga sentido en un teléfono, que sea de uso masivo, no específica, que solucione un problema cotidiano.

En este marco, una startup de Silicon Valley se propone desarrollar una plataforma genérica de sistemas de inventarios, que pueda ser instanciada desde la nube, para servir a múltiples instituciones que deseen hacer uso de la misma. Como esta startup cuenta con mucho capital de riesgo y pocas capacidades de programación ha decidido comisionar el diseño y desarrollo de este sistema, de nombre código "SAPo", a una pujante empresa Uruguaya de desarrollo de software.

Se busca que la plataforma a desarrollar cuente con las características clásicas de los sistemas de inventario sacando provecho de abstracciones y generalidades que se dan en todo sistema de estas características. Un usuario deberá definir lugares dentro del ciberespacio, llamemos *almacenes virtuales* (AV), los cuales en general se mapean con lugares en el mundo real, *almacenes reales* (AR). Por ejemplo un AV denominado Heladera se corresponde a la heladera de un hogar.

Un ser humano muchas veces no tiene acceso o no desea ir hasta el AR pero con este sistema tendrá acceso al mapeo (AV) en forma remota. Una aplicación que surge de esto sería llevar el control del stock de un almacén o el inventario del mobiliario de una empresa.

Debe soportar el ingreso de la información para poblar el AV de forma manual. Sin embargo la idea es que a futuro se pueda mediante una foto del AR (ej.: heladera) llegar a poblar el AV y sincronizarlo (con distintos dispositivos y plataformas). La principal tarea aquí pasa por identificar los elementos y cantidades de productos que hay en el AR para poblar el AV. El objetivo es que la aplicación sea capaz de identificar productos tal cual lo hace una máquina de un supermercado que utiliza una cajera.

La aplicación permite dar de baja lo que en la realidad equivaldría a consumir o descartar. Se puede editar la información existente mediante la toma de fotografías y edición de campos.

Concretamente y alineado a los objetivos se plantean las siguientes características:

- Administración del stock de productos hogareños, industriales, entre otros.
- Agregar fotografías de productos.
- “Llevarla a hacer las compras”
 - Poder dar de alta
 - Dada una lista de “qué tengo que comprar” sincronizar con lo que doy de alta
 - Me acuerdo de algo que hay que comprar o me avisan y se puede modificar quedando en todos los dispositivos sincronizado
 - Realizar web scrapping o integrarse con alguna API de un supermercado determinado para buscar o actualizar productos
- Algoritmo de recomendación de la lista de lo que tengo que comprar
- Utilizarla junto con la administración de los gastos
- Mover productos o listas de productos entre almacenes
- Compartir almacenes entre usuarios
- Chat y mensajes entre usuarios que comparten un almacén
- Comentarios a nivel de almacén que pueden ser utilizados como notas para sí mismo u otros usuarios
- Notificaciones

Dado que esta plataforma va a ser instanciada por instituciones que se dedican al control de inventario de productos completamente distintos, que pueden ir desde antigüedades, hasta cabezas de ganado, pasando por vehículos vintage y electrodomésticos es necesario que el sistema se pueda adaptar a los distintos tipos de datos que debe almacenar cada tipo de producto sin que esto involucre un cambio en el diseño, compilación, o incluso una nueva distribución del sistema. Por otro lado, teniendo en cuenta que se espera escalar agresivamente, el proceso de registro para una empresa que se propone montar un almacén virtual debe ser totalmente automático.

Como SAPo va a actuar como plataforma de base, sobre la cual van a montar su negocio virtual múltiples empresas, es de sumo interés que cada una de estas instanciaciones pueda contar con un alto grado de personalización en la interfaz de usuario. O sea, desde la perspectiva del usuario final, cada uno de estas instancias debe lucir como un sitio web diferente, alineado a la estética corporativa de la institución que lo está instanciando. Esto implica: potencialmente una

URL diferente, contenidos específicos, una estética distinta, sin verse afectado por los otros “inquilinos” de la plataforma.

El sistema deberá contar con dos módulos. Por un lado un *front office*, accesible mediante autenticación integrada con Facebook o con Twitter, permitiendo la administración del AV. A su vez deberá ser accesible como API de *web services* para ser invocado por ejemplo desde dispositivos móviles. Por otro lado un módulo de *back office* para administración general de la plataforma. Es decir que permita tanto la administración de almacenes específicos como las configuraciones del sistema (ej. integración, productos genéricos, entre otros).

Es deseable que se brinde soporte para la integración entre instalaciones de AV, permitiendo potenciar ciertas funcionalidades, tales como migración y asociación de AV entre instalaciones, búsqueda de productos entre almacenes, entre otros.

2. Requerimientos Funcionales

Módulo de Front Office

Las siguientes funcionalidades deben ser brindadas por el módulo de front office:

- Registro de usuarios del AV utilizando facebook o twitter
- Alta nuevos AV
- Página de inicio del usuario con:
 - Una lista de las notificaciones recientes
 - Una lista con los últimos movimientos de stock en los AVs
- Registro, baja o modificación de los datos o stock de un producto. La información del producto deberá ser flexible y por ejemplo podría incluir:
 - Nombre y descripción del producto
 - Un conjunto de etiquetas asociadas al producto que permita agrupar y facilitar las búsquedas
 - Un conjunto de imágenes del producto
 - Un precio de compra actual y su histórico
 - Un precio de venta actual y su histórico
 - Un stock actual y el histórico de los movimientos realizados
- Dar de alta productos y categorías específicas

- Notificaciones parametrizable cuando se cumple una condición para un producto determinado
- Pasaje a usuario pago¹
 - Ganar membresía mediante social media
 - Integración con las APIs de Paypal, o Google Checkout
- Reportes:
 - Valorización de los AVs según un atributo seleccionado por el usuario
 - Cantidad de movimientos por lapso

Módulo de Back Office

- Agregar administrador
- Administración de todos los AVs
 - Manejo de productos genéricos
 - Manejo de categorías de productos
- Envío de notificaciones asociadas a límites de la cuenta
- Reporte
 - Listado de usuarios registrados y movimientos realizados por usuario
 - Detección de fraude
 - Histórico de ganancias y proyecciones
 - Producto genérico más utilizado
- Promover producto (o categoría) específico a genérico sugerido automáticamente
- Proveer una interfaz Web mobile

Es obligatorio el despliegue en la nube (AWS u OpenShift) de la plataforma completa.

AVs de Ejemplo

Implementar al menos dos AVs de ejemplo sobre la plataforma construida que sirva para mostrar el uso de las funcionalidades que el sistema provee y de qué forma se utilizan.

3. Requerimientos Opcionales

1. Cliente Mobile (Android o iPhone) que permita mirar y modificar el inventario de los distintos AV que el usuario tenga acceso
2. Integración con Google Maps, o Bing Maps desde el modulo de back office, para visualizar los usuarios geográficamente
3. Integración entre grupos, permitiendo:
 - a. Listar almacenes y productos de otro sistema
 - b. Mover almacenes, productos o listas de productos entre sistemas

¹ La forma de monetizar la plataforma es responsabilidad del equipo implementador

4. Utilización de una base de datos NoSQL (por ej:MongoDB) para la persistencia de una parte de los datos del sistema
5. Automatizar la carga de productos de una categoría en base a la integración con sitios de terceras partes que mantengan catálogos. Por ejemplo *web scrapping* de sitios de tiendas similares a las que pueda referir el administrador

4. Agenda

14 de Septiembre - Inicio de Monitoreos y muestra de ejercicios prácticos

27 de Setiembre - Entrega arquitectura y diseño (por eva)

Semana 12 de Octubre - Presentación del prototipo de la Arquitectura.

26 de Octubre - Primera entrega del artículo.

Semana 2 de Noviembre Presentación del Sistema.

8 de Noviembre Entrega final.

Semana siguiente Presentaciones Finales.

5. Monitoreos

Los monitoreos serán realizados una vez por semana con una presentación del avance del trabajo al docente a cargo. Éste se encargará de realizar los comentarios pertinentes acerca del mismo. En los monitoreos con los grupos se utilizará el concepto de entregable. El mismo puede ser, de acuerdo al momento, una presentación PPT, un documento de avance o un prototipo del producto para revisar. En todos los monitoreos cada grupo debe traer un entregable, el cual se utilizará como base de discusión para esa reunión. Esto busca promover que los grupos estructuren y organicen sus consultas en torno al trabajo hecho en los días previos. El docente puede solicitar entregables adicionales a los mencionados en la Agenda.

6. Evaluación y Entrega Final

El curso se evalúa de acuerdo a las siguientes actividades:

- Entregas intermedias y participación en los monitoreos
- Documentación de la Entrega Final (particularmente el artículo)
- Producto desarrollado
- Presentación final (que incluye presentación del proyecto, solución y demo del producto).
- La entrega final que debe constar de:
 - **Software**
 - Código fuente de todas las funcionalidades y proyecto de desarrollo.
 - Scripts de ejecución instalación y ejecución automática de nombre ServidorX, (Donde X indica el orden de ejecución) incluyendo argumentos de línea de comandos o archivos de configuración si los tuviera.

- Archivo de nombre *readme* con una explicación rápida de parámetros u opciones que considere necesario aclarar. El archivo no debe contener más de una carilla.
- Opcional: Instalador de la aplicación
- Enlace del proyecto distribuido en la nube
- **Documentación**
 - Cronograma de desarrollo del obligatorio.
 - Documento de Arquitectura y Diseño de la solución.
 - Documento tipo artículo académico con la presentación de la solución, de entre 10 y 15 páginas.
 - Juego de datos de prueba.
 - Incluyendo información de ejecución.
 - Pequeño Video de Demo del Sistema.
 - Conjunto de Screenshots del Sistema.
 - Presentación.