

Proyecto Final de Carrera



SESE

**Sistema de Educación y Seguimiento
Escolar.**

Citroni Bruno –Pierini Maximiliano

Año: 2016

Director de Proyecto: Carlos Giorgetti.

Índice de contenidos

Índice de contenidos	2
1 Introducción	4
1.1 Objetivos técnicos	5
1.2 Objetivos profesionales	5
2 Análisis del Problema	6
2.1 Primer acercamiento con el cliente y definición del problema	6
2.2 Requisitos del Cliente	7
3 Soluciones propuestas	9
3.1 Tecnologías de desarrollo	9
3.2 Arquitectura de Software	11
3.2.1 Modelo - Vista – Controlador.	12
3.3 Soluciones Determinadas a Problemas específicos	14
3.3.1 Gráficos para representación de estadísticas.	14
3.3.2 Delimitación de los barrios a través de Google Maps.	16
4 Metodología Ágil Utilizada: Scrum	17
4.1 Definición y Principios	18
4.2 Roles	19
4.3 Ceremonias	20
4.3.1 Sprint Planning Meeting:	20
4.3.2 DailyScrum Meetings:	20
4.3.3 Sprint Review& Sprint Retrospective:	21
4.4 Elementos	21
4.4.1 Poda de Requerimientos	21
4.4.2 ProductBacklog	21
5 Herramientas Utilizadas.	23
5.1 Aptana Studio 3.	23
5.2 Tortoise SVN	23
5.3 WAMP Server	24
5.4 MySQLWorkbench	24

5.5	Cliente de base de datos. Navicat.	25
5.6	Herramientas de Firefox. Firebug.	25
6	SprintBacklog	26
6.1	Sprint 1	26
6.2	Sprint 2	28
6.3	Sprint 3	31
6.4	Sprint 4	33
6.5	Sprint 5	36
7	Migración de Datos	38
8	Pruebas de software	40
8.1	Scrum y Testing	40
8.1.1	Rol del Tester	40
8.2	Pruebas aplicadas al sistema.	41
8.2.1	Pruebas de unidad.	41
8.2.2	Pruebas de integración.	42
8.2.3	Pruebas funcionales.	43
9	Plan de Implementación	44
10	Conclusión	45
11	Bibliografía.	46
12	Anexo 1: Diseño Web Responsivo o Responsive WebDesign.	47
	Primer paso: El diseño fluido	48
	Segundo paso: Los Media Queries	50
	¿Qué navegadores soportan los Media Query?	51
13	Anexo 2: Diseño final de la interfaz implementada en el sistema de educación.	52
14	Anexo 3. Visualización y Delimitación Geográfica de los Barrios a Través de Google Maps	63
15	ANEXO 4. Instalación del software en entorno Windows Server	67
16	Anexo 5. DER del sistema de educación y seguimiento escolar para el movimiento “Los Sin Techo”.	69
17	ANEXO 6. Diagrama de despliegue.	70

1 Introducción

El Movimiento Los Sin Techo es una organización no gubernamental que trabaja para el desarrollo integral y la organización comunitaria del sector marginado de la ciudad de Santa Fe, Argentina. Desde 1985 ha desarrollado distintas iniciativas tendientes a encontrar respuestas a los problemas estructurales de los más pobres.

Desde el año 2000 viene desarrollando una segunda etapa destinada a proteger el derecho a la vida y ayudar a los niños a alcanzar niveles de desarrollo intelectual normales.

Ha instalado la primera red inalámbrica de Internet en los barrios periféricos de Santa Fe con más de 120 computadoras destinadas a la estimulación educativa de niños, jóvenes y adultos y a su integración a la sociedad y al conocimiento.

Su fundador y director, durante gran parte de su vida, el padre Atilio Rosso, fue el principal impulsor de la incorporación de la educación dentro del plan integral del movimiento. Dicha convicción sobre la educación en los sectores marginales de nuestra sociedad llevó al movimiento a crear distintos departamentos educativos, cada uno de ellos dirigido a distintos grupos etarios.

El sistema de educación y seguimiento escolar "SESE", fue principalmente dirigido a satisfacer las necesidades de los departamentos "JARIDNES" y "PRIMERO MI PRIMARIA".

La implementación de este aplicativo se basa en la necesidad de contar con un sistema de información en tiempo real y con procesos continuos y sistemáticos, que asistan en forma permanente a la toma de decisiones que involucren al sector educativo.

La integración de los distintos módulos educativos en un único sistema de información reduce los tiempos requeridos para la obtención y procesamiento de la información y mejora la calidad e integridad de la información relevada.

El movimiento "Los Sin Techo" han decidido llevar adelante un plan de seguimiento escolar y actualización educativa, en base al cual se pensó en la oportunidad de tener un sistema de información integral capaz de brindarles las siguientes mejoras al proceso actual de seguimiento del sector educativo:

- Obtención de información sobre los distintos factores educativos.
- Generación de informes y estadísticas en base a la información recolectada.
- Eficiencia en el manejo de los datos y disponibilidad sobre los mismos.
- Rápida toma de decisiones.
- Situación del sector educativo en tiempo real.
- Capacidad de seguimiento de alumnos dentro de los distintos módulos educativos.
- Definición de procesos de evaluación para los distintos sectores.
- Seguimiento del ambiente en el cual los alumnos desarrollan sus vidas.
- Disponibilidad de la información.

1.1 Objetivos técnicos

El objetivo principal de este proyecto es brindar una mejora sustancial al proceso educativo del movimiento “Los Sin Techo”. Para alcanzar dicha meta pensamos en el desarrollo de un sistema de información que permita el manejo de todos aquellos datos que son de vital importancia, no solo para el seguimiento de los alumnos incluidos dentro del proyecto, sino también para lograr una mejora sostenida en el proceso de toma de decisiones. Esto repercutirá en el futuro de los alumnos, favoreciendo la consolidación del modelo educativo, propuesto por el movimiento, como herramienta social para disminuir la exclusión y marginalidad de los sectores más carenciados.

Pretendemos generar un software confiable y robusto, que pueda operar en diferentes plataformas y por sobre todas las cosas que sea fácilmente mantenible. Todo esto sin dejar de lado la usabilidad del sistema, pieza clave del sistema “SESE”, ya que el uso del mismo está dirigido no solo a las autoridades coordinadoras del sector educativo del movimiento, sino que también tendrá como participe principal a las madres que asisten y ayudan en el desempeño de las distintas tareas vinculadas al área educativa.

Para poder respetar la premisa fundamental del sistema que es la usabilidad del mismo, dentro del anexo número 2 - “Diseño final de la interfaz implementada en el sistema de educación”- se podrán observar las interfaces principales del sistema de educación y seguimiento escolar. Las mismas fueron pensadas para garantizar la sencillez tanto de interpretación como de uso.

1.2 Objetivos profesionales

En cuanto al desarrollo como profesionales planificamos no solo experimentar con metodologías ágiles para la gestión de proyecto (situación con la que aún no hemos convivido en un entorno real), sino también aprovechar las ventajas de dicha metodología para optimizar todos los procesos asociados al desarrollo del sistema de información.

Dicha optimización nos permitirá mejorar la capacidad de relevamiento de requisitos, haciendo foco principalmente en la comunicación informal con los usuarios a los cuales está orientado dicho sistema, y aprovechando la predisposición de los mismos para lograr una retroalimentación constante que nos permita tomar las mejores decisiones desde la perspectiva del desarrollo de sistemas.

Esta experiencia nos dará la oportunidad de incorporar nuevos conocimientos asociados a distintas herramientas de desarrollo. Todo lo aprendido nos hará capaces de exprimir al máximo todos sus beneficios, en pro de obtener un sistema de información de mejor calidad y en consonancia con los requerimientos tecnológicos actuales.

Un aspecto fundamental que debemos resaltar en este punto es la posibilidad que se nos presenta como profesionales de poder aplicar nuestros conocimientos en pos de brindar un servicio a la sociedad y de esta forma contribuir a la mejora del ambiente social en el cual estamos inmersos.

2 Análisis del Problema

2.1 **Primer acercamiento con el cliente y definición del problema**

Todo el proyecto tiene como punto de partida nuestra participación dentro de un proyecto de voluntariado universitario de nuestra casa de estudios. Dichos voluntariados tienen como objetivo dar a los alumnos de la universidad la posibilidad de comenzar a aplicar sus conocimientos en pos de contribuir a distintos aspectos sociales que se presentan en el ambiente que rodea a nuestra facultad.

En base a lo mencionado anteriormente el primer acercamiento con “el cliente” se llevó a cabo en nuestra facultad, donde se realizó una presentación de la ONG y luego se fueron presentando los distintos temas que revestían mayor importancia y que como resultado de dicha importancia debían ser abordados, desde el ámbito de los sistemas de información, en forma inmediata.

En esta reunión, se nos manifestó el deseo por parte del movimiento “Los Sin Techo” de poder contar con un sistema de información que permita la gestión integral del sector educativo, abarcando las distintas ramas que este posee en la actualidad.

Una vez enunciado el objetivo principal del sistema a implementar, comenzamos a tomar nota sobre cuáles eran los principales inconvenientes que se presentaban en la actualidad, para de esta forma poder tener una base firme que nos permita mejorar y optimizar la etapa de captura de requisitos. El usuario comenzó mencionando que actualmente disponen de una serie de planillas en formato Excel, las cuales utilizaban para cargar la información obtenida de los distintos niveles educativos, es decir que dichas tablas eran utilizadas como una especie de base de datos de relativa sencillez. Uno de los principales inconvenientes que presenta la utilización de dichas planillas como base de datos es que dificulta el manejo de grandes caudales de información y la utilización de los mismos para la obtención de estadísticas que permitan medir el rendimiento de los alumnos y de las metodologías educativas aplicadas para tal fin.

Además se nos mencionó que dicha base de datos presentaba datos dispersos, quizás repetidos y no consolidados que llevaban en muchos casos a cometer distintos errores de relativa importancia en el cálculo de las distintas estadísticas. Este tipo de inconvenientes imposibilitaban muchas veces la obtención de reportes capaces de reflejar la situación actual del módulo educativo de la ONG.

Luego de comprender cuáles eran los principales problemas actuales, el cliente comenzó con una sencilla descripción de cómo es la estructura organizacional, quienes son los responsables de cada área educativa, y a modo de anticipo nos fue enunciado los resultados esperados que se desprendan de la utilización del sistema de información. Dichos resultados no solo reflejan importantes cambios en cada módulo educativo sino que también sirven como base a los dirigentes del movimiento para poder saber si están transitando el camino correcto desde la perspectiva educacional.

En todo momento el cliente nos reflejó la necesidad de que el sistema de información posea interfaces de usuarios sencillas y muy intuitivas, ya que el mismo iba a ser utilizado por personas que quizás no tienen la posibilidad de contar con muchos conocimientos del sector informático y

por lo tanto quizás no sean capaces de poder comprender funcionalidades que posean una lógica complicada.

Como punto final de la reunión, por un lado se nos informó que tenían la intención de poder ir implementando el sistema en forma escalonada en cada área educativa, para que de esta forma se pueda hacer un seguimiento más exhaustivo de la puesta en "producción" del sistema "SESE". Como segundo punto a destacar se nos invitó a realizar una pequeña visita a distintos lugares que posee el movimiento, para de esta forma poder sumergirnos en la realidad con la que ellos interactúan día tras día.

2.2 Requisitos del Cliente

Como resultado del primer acercamiento logramos confeccionar una lista con los requisitos principales del cliente y que nos servirán como punto de partida para poder comenzar con el análisis de los mismos.

A continuación se enuncian los requisitos mencionados anteriormente:

- Mantener un listado centralizado de alumnos que participan en los distintos módulos educativos ofrecidos por el movimiento.
- Realizar altas, bajas, modificaciones tanto de alumnos, barrios, centros educativos, grupos, etc. Se debe poder realizar en forma rápida, segura y a través de una interfaz sumamente amigable.
- Contar con la posibilidad de gestionar las asistencias de los alumnos a cada una de las actividades, para de esta forma poder detectar cualquier problema que se presente y que imposibilite al chico asistir regularmente a clases.
- Poseer una metodología rápida y sencilla de carga de datos, ya que muchas veces estas tareas serán realizadas por personas que no poseen muchos conocimientos sobre informática.
- Migración hacia el nuevo sistema de toda la información con la que el movimiento cuenta hasta el momento.
- Permitir la obtención de algunas estadísticas predefinidas que incluyan gráficos representativos de las mismas. Estas se tomarían como base para poder controlar el funcionamiento de las políticas educativas que son implementadas en cada sector educativo.
- Contar con la posibilidad de buscar o filtrar aquella información que los usuarios deseen en forma rápida, dependiendo del menú donde el mismo se encuentre posicionado y que de esta forma le permitiera encontrar sin demoras cualquier dato con el que desee trabajar o simplemente consultar.
- Poder contar con un módulo de evaluaciones, donde el mismo sea utilizado para la carga de los resultados de las mismas, sobre las dos instancias evaluativas con las que se cuenta en la actualidad (parcial o final).
- Contar con una gestión de usuarios que permita la participación de distintos perfiles y que sobre estos se presenten reglas para definir los permisos asociados a cada perfil establecido.

Como resultado del acercamiento preliminar realizado con el cliente y del listado general de requisitos mencionados anteriormente, procedimos a confeccionar un listado con requerimientos funcionales para nuestro sistema de información. También hemos decidido agregar un listado de requerimientos no funcionales en los cuales nos enfocáramos para garantizar la calidad y el valor agregado de nuestro sistema. Dicha lista fue confeccionada no solo con requerimientos que partieron desde nuestro lado, sino también con aquellos que se desprendieron del lado del cliente, y que el mismo hizo mención como primordiales o de mayor relevancia.

Requisitos funcionales:

Nombre Funcionalidad	Descripción General
Gestionar Alumnos	El usuario podrá administrar toda la información relacionado con los alumnos participantes de los distintos módulos educativos.
Gestionar Barrios	El usuario podrá crear, modificar y consultar toda la información relacionada con los barrios donde se puedan encontrar centros pertenecientes al movimiento.
Gestionar Centros Educativos	El usuario podrá crear, modificar y consultar los centros educativos que posee la ONG.
Gestionar Usuarios	El usuario podrá crear, modificar y consultar los usuarios y sus perfiles asociados.
Gestionar Asistencias	El usuario podrá cargar y obtener información sobre las asistencias de cada alumno a cada actividad en la cual esté inscripto.
Gestionar Grupos	El usuario podrá crear, consultar y modificar los grupos con los que cuenta cada centro educativo del movimiento.
Gestionar Estadísticas	El usuario podrá generar estadísticas en base a la información cargada con anterioridad en el sistema.
Gestionar Evaluaciones	El usuario podrá cargar los resultados de las evaluaciones que realice en cada grupo.
Gestionar Avisos	El usuario podrá crear avisos para que los demás usuarios puedan estar al tanto de la última información importante.
Importar documentos a Excel y PDF	El usuario podrá imprimir o visualizar información filtrada o no de la base de datos del sistema, esta información se presentara como un documento en Excel o Pdf como así lo requiera.

Requisitos No funcionales:

Requisito	Descripción General
Usabilidad	El sistema deberá presentar una interfaz muy amigable y sobre todo muy intuitiva.
Mantenibilidad	El sistema deberá contar con un nivel de mantenimiento básico y muy sencillo para que pueda ser realizado en primera instancia por el cliente.
Disponibilidad	El sistema deberá encontrarse funcional el 99.9% del tiempo en que sus usuarios deseen utilizarlo.
Modificabilidad	El sistema deberá adecuarse a los cambios fácilmente.
Portabilidad	El sistema deberá poder correr tanto en los equipos destinados para su desarrollo como en todos los equipos del cliente.

Integrabilidad	El sistema podrá ser desarrollado como componentes separados, que al juntarlos, trabajaran correctamente.
----------------	---

3 Soluciones propuestas

Este conjunto de soluciones, tecnologías, metodología y arquitectura para el sistema fueron evaluadas durante un lapso de 30 días como actividad previa al comienzo del primer sprint.

3.1 Tecnologías de desarrollo

Este proyecto es encarado desde los conocimientos que posee cada uno de los integrantes del equipo. En base a esto se han realizado distintas comparaciones que nos fueron guiando hasta alcanzar las definiciones finales de distintos aspectos vinculados con el desarrollo del sistema de información. La mayoría de las tecnologías utilizadas fueron las que a nuestro criterio eran las más adaptables a este tipo de desarrollo.

Para encarar este proyecto utilizamos PHP como lenguaje de programación:

PHP es un lenguaje de programación, de uso general y diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

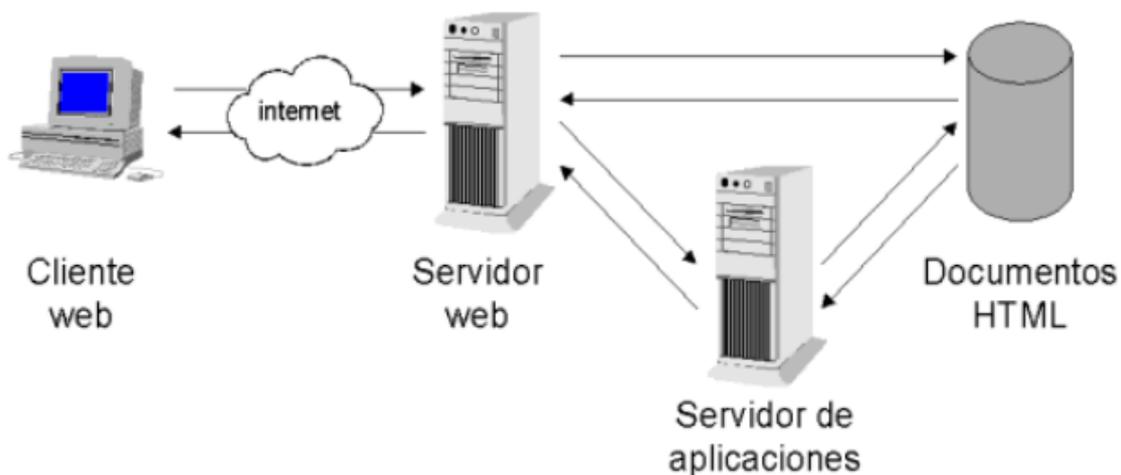


Figura 1-Estructura Cliente-Servidor.

A continuación se muestra un ejemplo de cómo funciona PHP a la hora de ejecutar alguna sentencia de código expresada en dicho lenguaje de programación.

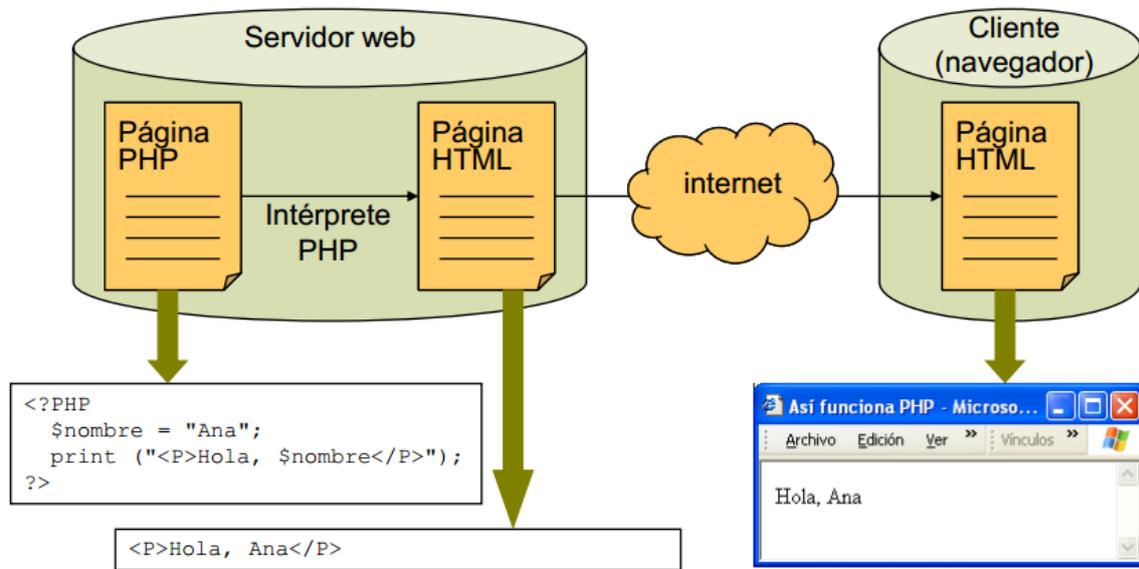


Figura 2-Ejemplo de Funcionamiento Básico de PHP.

Características del lenguaje PHP.

Ventajas:

- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los

desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Desventajas:

- Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución.

3.2 Arquitectura de Software

La siguiente etapa a iniciar consistió en la definición de la arquitectura que debíamos utilizar para nuestro sistema de información.

Para llevar adelante esta tarea, de la mejor forma posible, nos propusimos realizar una lista de las características que debería cumplir el sistema a nivel arquitectónico, las cuales nos servirían luego a la hora de inclinarnos hacia alguna arquitectura en particular. Dichas características son las que a continuación se detallan:

- Era necesario que los datos estén **almacenados de forma centralizada** puesto que la información sería consultada desde cualquiera de los puntos desde el cual se tenga acceso al sistema de información.
- Un punto importante a considerar es que la **interfaz del usuario tenderá a cambiar constantemente** a medida que el cliente realice pruebas sobre cada incremento del sistema.
- La **carga de información** realizada en el sistema deben realizarse **con total seguridad** para que las estadísticas obtenidas sobre dichos datos sea lo más verídica posibles y sobre todas las cosas mantengan un alto grado de representatividad.
- El sistema deberá tener un **alto grado de disponibilidad** para no afectar la toma de decisiones que se puede llegar a desprender de la información obtenida mediante la consulta del sistema.

En base a las características anteriormente mencionadas nos dispusimos a realizar una evaluación de los diferentes tipos de arquitecturas disponibles cuya factibilidad de uso sea de un alto grado. Tomando como base el hecho que debíamos respetar las consideraciones mencionadas más arriba, llegamos a la conclusión de que una arquitectura en capas sería la más adecuada para nuestro sistema en particular. La definición clásica de esta arquitectura está definida por una capa de presentación donde corren las interfaces, una de control en la cual se ejecuta la lógica del negocio y una capa de datos encargada de la persistencia de los objetos.

3.2.1 Modelo - Vista - Controlador.

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

El usuario interactúa con dicha arquitectura de la siguiente forma:

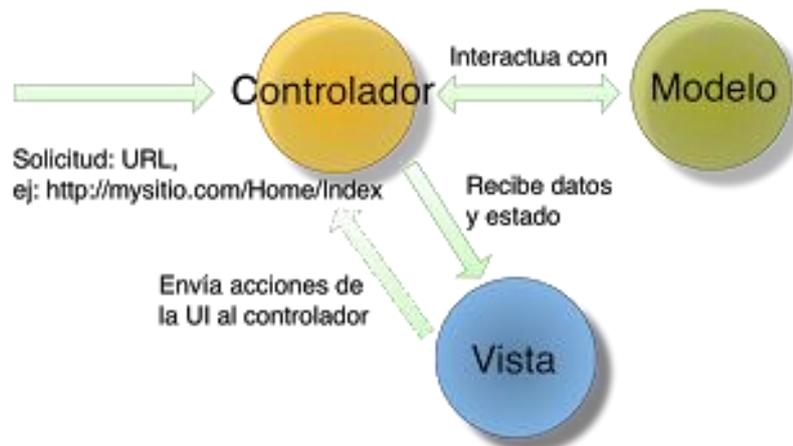


Figura 3-Esquema interacción entre el Usuario y la arquitectura.

3.2.1.1 Capas de la aplicación

Cuando se utiliza el patrón MVC, una aplicación se divide en las siguientes capas:

El modelo

Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

La vista

Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

El controlador

Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un

documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo' .
Con el siguiente cuadro se pretende ejemplificar como es la interacción entre cada uno de los componentes que conforman este modelo arquitectónico.

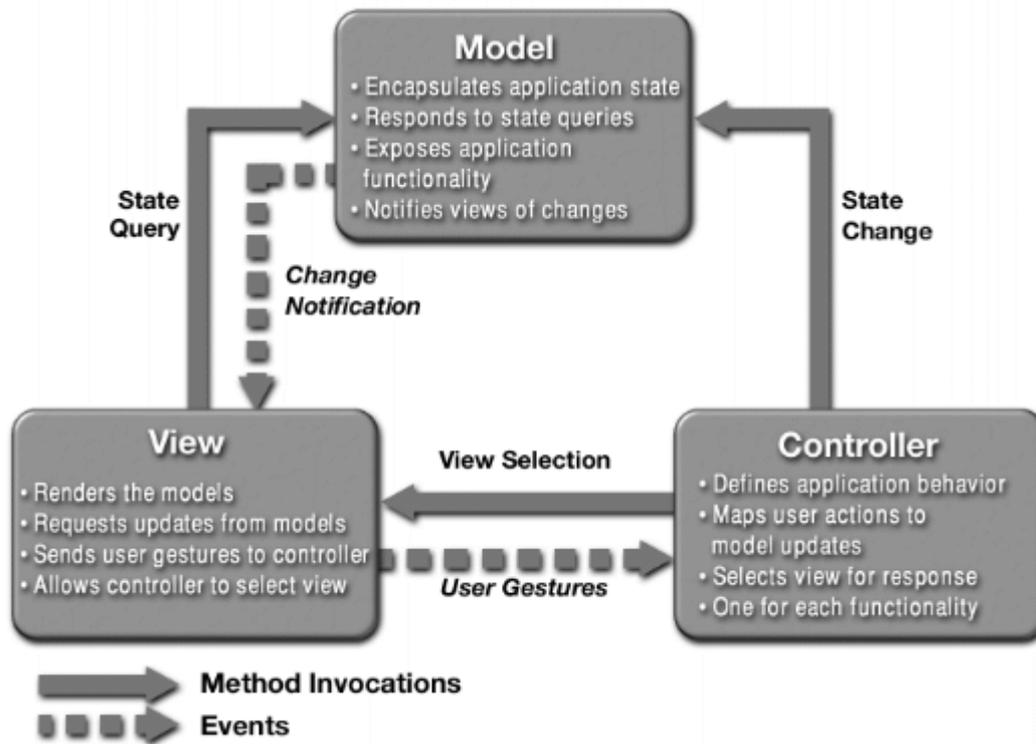


Figura 4-Intereacción entre los diferentes componentes de la arquitectura.

3.2.1.2 Ventajas de las capas

Las ventajas asociadas a la utilización de esta arquitectura para el desarrollo de software son muchas y cada una de ellas persigue la optimización de distintos componentes asociados al sistema en cuestión.

Algunas de ellas son:

- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables.
- Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.

- MVC está demostrando ser un patrón de diseño bien elaborado puesto que las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

3.3 Soluciones Determinadas a Problemas específicos

A medida que íbamos superando cada una de las etapas asociadas al desarrollo del sistema, se nos fueron presentando diferentes problemas asociados a circunstancias específicas relacionadas a distintas funcionalidades del sistema en conjunto.

Dentro de estos problemas/desafíos que se nos fueron presentando, podemos mencionar dos que tuvieron una importancia mayor al momento de analizar su posible solución y que directa o indirectamente afectaban algunas de las funcionalidades más importantes de nuestro sistema.

A continuación se hará mención del problema y de la solución que decidimos adoptar para afrontar las circunstancias que el mismo nos presentaba.

3.3.1 Gráficos para representación de estadísticas.

Desde un primer momento se nos planteó desde el movimiento la posibilidad de contar con estadísticas que sean los más representativas posibles de cada uno de los sectores educativos que comprenden a la organización.

En un primer momento realizamos un análisis para poder determinar cuáles serían aquellas estadísticas que le permitan al movimiento contar con información importante al momento de tomar decisiones o simplemente para realizar un análisis del funcionamiento de los planes educativos actuales.

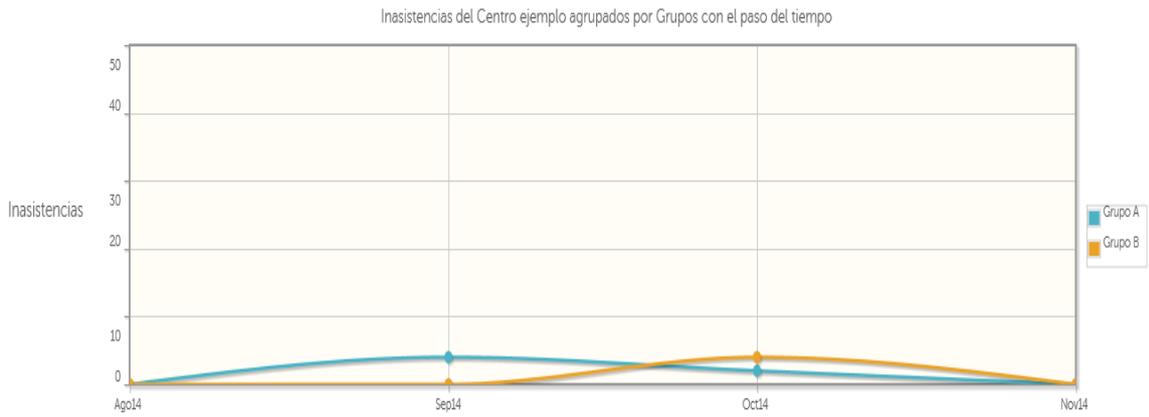
Más allá de las estadísticas analizadas anteriormente pensamos en una forma de representación de las mismas, que sea capaz de trasladar la parte numérica de las mismas a gráficos que sean sencillos de leer y por sobre todas las cosas que puedan ser de vital ayuda para la toma de decisiones por parte de la dirigencia del movimiento.

Para poder resolver este desafío haremos uso de una herramienta que nos permite realizar distintos tipos de gráficos estadísticos en base a la información cargada en el sistema. Dicha herramienta es la llamada "JQPLOT".

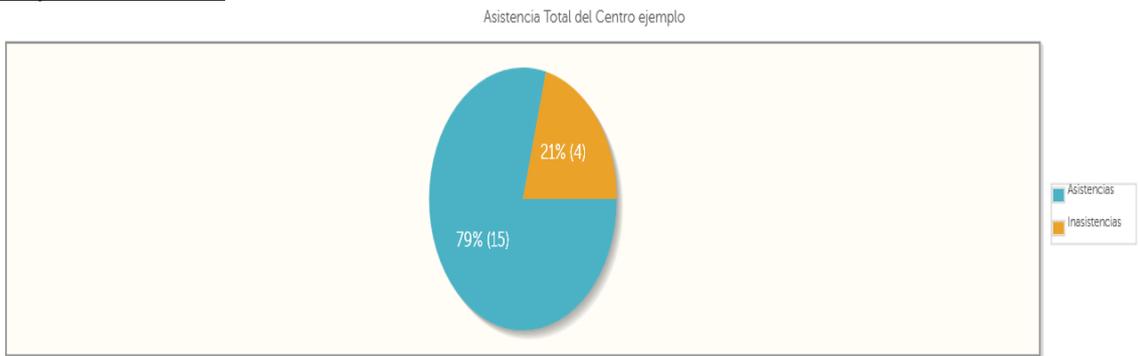
JQPLOT se trata de un *plugin* jQuery que genera gráficos estadísticos en el navegador dentro de las páginas web donde sea necesario. Soporta gráficos lineales, de barras y circulares (en forma de torta). Tiene numerosas opciones de configuración como la inclusión de sombras y la interacción con eventos *drag&drop*. JQPLOT ha sido probado en IE7, IE8 o superior, Firefox, Chrome, Safari y Opera. La licencia de uso es MIT y GPL versión 2, por lo que puede ser utilizado libremente y en forma gratuita dentro del proyecto. Para su correcto funcionamiento el mismo requiere jQuery (1.4.3 o superior para algunas características).

A continuación se muestran algunas imágenes con diferentes tipos de gráficos que se pueden obtener mediante la utilización de dicha herramienta.

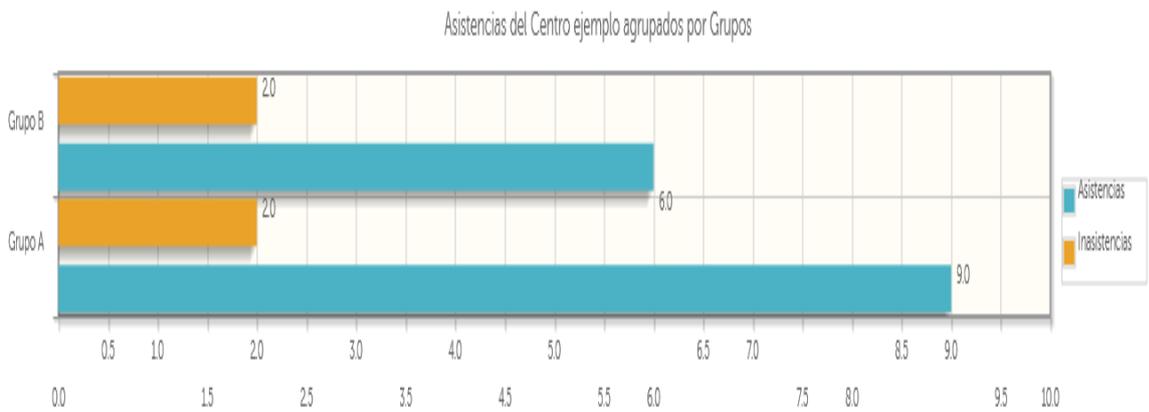
Graficas lineales:



Graficas circulares:



Graficas de barras:



3.3.2 Delimitación de los barrios a través de Google Maps.

He aquí otro punto que debemos remarcar en cuanto a consideraciones particulares que se debieron realizar a medida que el desarrollo del sistema de educación iba avanzando. Teniendo en cuenta cuáles serían los usuarios finales de dicho sistema y cuál sería el uso que cada uno de ellos podía realizar del mismo, vimos como una muy buena posibilidad la incorporación de la delimitación de los barrios en los que el movimiento cuenta con una amplia participación.

Desde los representantes del área educativa del movimiento surgió la idea de que aquellos usuarios del sistema, que trabajen en cada uno de los barrios, puedan observar en una forma representativa los límites del barrio donde ellos tienen injerencia. Dicha información les posibilitaría organizar de una forma mucho más eficiente los grupos, logrando una mejor optimización de los recursos que el movimiento posee en sus distintas áreas.

Desde el punto de vista del desarrollo pensamos en la posibilidad de incorporar, mediante el uso de distintas herramientas, una imagen que represente de la mejor forma posible la delimitación del barrio. Para poder realizar dicha delimitación procedimos a utilizar la información que fue recolectada en una práctica profesional supervisada, la cual nos permitió la obtención de los puntos representativos de los polígonos asociados a cada barrio.

Al momento de analizar cuál sería la mejor herramienta que podríamos utilizar para poder representar gráficamente los barrios, y luego de realizar una etapa de búsqueda de información asociada a esta, nos decidimos por "Google Maps".

Google Maps es un servidor de aplicaciones de mapas en la web que pertenece a Google. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calles.

Google Maps ofrece la capacidad de realizar acercamientos y alejamientos para mostrar el mapa. El usuario puede controlar el mapa con el mouse o las teclas de dirección para moverse a la ubicación que se desee. Para permitir un movimiento más rápido, las teclas "+" y "-" pueden ser usadas para controlar el nivel de zoom. Los usuarios pueden ingresar una dirección, una intersección o un área en general para buscar en el mapa.

GoogleMaps provee a los desarrolladores un API capaz de aprovechar los datos disponibles a través del servicio, en el seno de las propias aplicaciones. De esta forma pudimos hacer uso de la misma para poder, mediante los puntos delimitantes de los distintos barrios, obtener una imagen lo más certera posible de cada uno de los barrios donde el movimiento cuenta con algún tipo de participación social.

Todos los beneficios y facilidades que nos brinda Google Maps para nuestro sistema en particular están ejemplificados en el anexo número 3 – "Visualización y Delimitación Geográfica de los Barrios a Través de Google Maps". Dentro del mismo se podrán visualizar algunos

ejemplos de la forma en la cual se implementó Google Maps para poder realizar la delimitación de los barrios que están incluidos dentro del radio de acción del movimiento.

4 **Metodología Ágil Utilizada: Scrum**

Como en todo proceso de desarrollo de software debemos seleccionar una metodología de desarrollo que se adapte de la mejor forma posible al ámbito y a las necesidades propias del sistema de información que vayamos a crear. Dicha decisión la realizamos teniendo como base el conocimiento que poseíamos de cada una de las metodologías de desarrollo más comunes, y partiendo de este punto nuestra idea fue la de poder implementar una metodología ágil. Este proyecto también nos sirvió como una forma de experimentar como es el desarrollo bajo este tipo de metodología ya que nuestra experiencia con la misma era escasa.

El desarrollo ágil de software es un conjunto de métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requerimientos y soluciones evolucionan mediante la colaboración de grupos auto-organizados y multidisciplinarios. Existen muchos métodos de desarrollo ágil, la mayoría minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, sino que la meta es tener una «demo» (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.

Desde el punto de vista propio del desarrollo también se decidió utilizar este tipo de metodología ya que veíamos prioritaria la participación activa del cliente, para este caso en particular la gente vinculada con el movimiento "Los Sin Techo". Esta constante interacción con los responsables de los distintos sectores educativos de la organización hizo que la idea de utilizar este tipo de metodología fuera altamente aplicable y la utilización de la misma representaría sin lugar a dudas la posibilidad de aprovechar al máximo la participación del cliente y de esta forma alcanzar un producto capaz de satisfacer desde todos los aspectos las distintas expectativas que se fueron generando reunión tras reunión.

4.1 Definición y Principios

Scrum es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Reduce al máximo la burocracia y actividades no orientadas a producir software que funcione y produce resultados en periodos muy breves de tiempo.

Podemos ver a **Scrum** como un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.

Como método, **Scrum** enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, prácticas de desarrollo, implementación y demás cuestiones técnicas. Más bien delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posible.

Se basa en los principios ágiles:

- Privilegiar el valor de la gente sobre el valor de los procesos.
- Entregar software funcional lo más pronto posible.
- Predisposición y respuesta al cambio.
- Fortalecer la comunicación y la colaboración.
- Comunicación verbal directa entre los implicados en el proyecto.
- Simplicidad; supresión de artefactos innecesarios en la gestión del proyecto.



Figura 5-Ciclo de vida de la metodología Scrum.

4.2 Roles

La dimensión del equipo total de **Scrum** no debería ser superior a veinte personas. Si hay más, lo más recomendable es formar varios equipos. No hay una técnica oficial para coordinar equipos múltiples, pero se han documentado experiencias de hasta 800 miembros, divididos en **Scrums** de **Scrum**, definiendo un equipo central que se encarga de la coordinación, las pruebas cruzadas y la rotación de los miembros.

Scrum tiene una estructura muy simple. Todas las responsabilidades del proyecto se reparten en 3 roles principales:

- **Productowner o Dueño del producto:** Este representa a todos los interesados en el producto final. Posee responsabilidades en las áreas de financiación del proyecto, en los requisitos del sistema, en el retorno de la inversión del proyecto y en su lanzamiento.

Es el responsable oficial del proyecto, de su gestión, control y de la visibilidad de la lista de acumulación o lista de retraso del producto (ProductBacklog). Toma las decisiones finales de las tareas asignadas al registro y convierte sus elementos en rasgos a desarrollar.

Rol Asumido por: Movimiento “Los Sin Techo”, Citroni Bruno, Pierini Maximiliano.

- **Scrum Master o Líder del proyecto:** Este es el responsable del proceso **Scrum**, de cumplir la meta y resolver los problemas. Así como también, de asegurarse que el proyecto se lleve a cabo de acuerdo con las prácticas, valores y reglas de **Scrum** y que progrese según lo previsto. Interactúa con el cliente y el equipo. Coordina los encuentros diarios, y se encarga de eliminar eventuales obstáculos. Debe ser miembro del equipo y trabajar a la par.

Rol Asumido por: Citroni Bruno, Pierini Maximiliano.

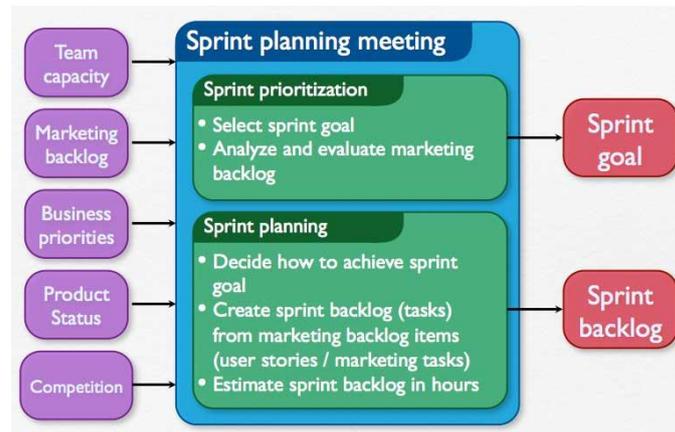
- **Team o Equipo:** Responsable de transformar el Backlog de la iteración en un incremento de la funcionalidad del software. Tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos. Se caracteriza por ser Auto-gestionado, Auto-organizado y Multi-funcional.

Rol Asumido por: Citroni Bruno, Pierini Maximiliano.

4.3 Ceremonias

4.3.1 Sprint Planning Meeting:

Se lleva a cabo al inicio del sprint y su duración es de 8 horas (máximo). Se divide en dos partes. En la primera se define el alcance del sprint definiendo el sprint goal y en la segunda se definen las tareas para el sprint que se reflejan en el Sprint Backlog, detallando el tiempo que tomará realizar cada una de ellas.



6-Entradas y salidas de una Sprint Planning Meeting.

4.3.2 DailyScrum Meetings:

Principales características:

- Reunión diaria,
- Duración: 15 minutos,
- No se trata de una reunión para resolver problemas,
- Se la conoce también como “Daily Stand Up”: debe realizarse de pie para respetar la duración de la misma y para que se desarrollen únicamente las 3 preguntas indicadas a continuación.

Debemos encontrar respuestas para las siguientes preguntas:

- ¿Qué hiciste ayer?
- ¿Qué harás mañana?
- ¿Qué impedimentos hay en tus tareas?

Muchas veces nos podemos ver tentados por analizar la posibilidad de reemplazar este tipo de reuniones por otro medio de comunicación. La idea de contar con estas reuniones es que el equipo entero pueda visualizar el estado del sprint cada día y crear presión de pares.

4.3.3 Sprint Review & Sprint Retrospective:

Al final del ciclo sprint, dos reuniones se llevarán a cabo: la “Reunión de Revisión del Sprint” y la “Retrospectiva del Sprint”.

4.3.3.1 Sprint Review Meeting:

- Los equipos presentan lo que realizaron durante el sprint,
- Típicamente tiene la forma de una Demo,
- Informal: 2 horas de preparación es la regla,
- Participantes: Clientes, Gerentes, ProductOwner, Otros Ingenieros.

4.3.3.2 Sprint Retrospective Meeting:

- Sólo se reúne el ScrumTeam,
- Es una reunión para obtener feedback
- Cada miembro del equipo presenta su visión sobre:
 - ✓ Qué estuvo OK y que no,
 - ✓ Qué puede ser mejorado,
 - ✓ Cómo implementar las mejoras.

4.4 Elementos

4.4.1 Poda de Requerimientos

La primera actividad es armar una lista exhaustiva de los requerimientos originales del sistema. Luego se procede a ver qué requerimientos son realmente necesarios, cuáles pueden posponerse y cuáles eliminarse.

Para esto debe poder identificarse un representante que posea capacidad para la toma de decisiones, priorizar los requerimientos en base a su importancia y acordar cuáles son los prioritarios para la fecha de entrega.

La poda de requerimientos es una buena práctica implícita en modelos ágiles, se hace lo que el cliente realmente desea, no más.

Para nuestro proyecto en particular esta práctica nos fue de suma utilidad para poder realizar un sistema acorde a lo que el cliente esperaba y deseaba, pero siempre dejando abierta la posibilidad de futuras mejoras sobre el mismo.

4.4.2 ProductBacklog

Esta sección nos permitirá presentar la lista de producto final que contiene todas las historias desarrolladas durante el proyecto.

Las estimaciones fueron realizadas en StoryPoints (SP), donde: $1\text{ SP} \cong 8\text{ horas} - \text{hombre}$; y las prioridades se encuentran dentro del intervalo [1-16] donde 1 representa la prioridad más alta.

Id.	Historia	Prioridad	Estimación	Sprint
0	Establecer el ProductBacklog Inicial del Sistema	1	4	1
1	Diseño de interfaces principales	1	7	1
2	Creación Del Modelo de Datos	1	2	1
13	Gestionar Asistencias	2	6	1
15	Gestionar Alumnos	2	4	1
3	Creación Plan de Pruebas	3	1	1
4	Diseño de interfaces del sistema	3	12	2
5	Diseño del menú principal	3	2	2
7	Diseñar interfaz de login	3	1	2
17	Definir permisos de usuarios	4	1	2
12	Gestionar Usuarios	5	5	2
24	Definir políticas de seguridad	5	1	2
8	Gestionar Barrios	6	7	3
9	Gestionar Centro de Educación	6	3	3
10	Gestionar Grupos	6	5	3
6	Poblar BD con datos	8	4	3
21	Gestionar evaluaciones	6	6	4
22	Gestionar estadísticas asistencias	8	7	4
23	Gestionar estadísticas evaluaciones	8	7	4
16	Gestionar avisos	9	2	4
34	Definir representaciones graficas sobre estadísticas	9	4	4
25	Cerrar listado de Requerimientos	9	1	5
27	Crear Reportes	9	5	5
33	Diseñar interfaces adaptables	10	5	5
14	Refinar Diseño de Interfaces	11	2	5
29	Probar el Sistema	11	5	5
26	Aplicar políticas de mejoras de rendimiento	12	2	5
31	Preparar la instalación	13	3	6
28	Instalar aplicación en el servidor	14	5	6
30	Preparar plan de capacitación	15	3	6
32	Brindar capacitación a los usuarios	15	5	6

5 Herramientas Utilizadas.

5.1 **Aptana Studio 3.**

Aptana Studio es un entorno de desarrollo integrado de software libre basado en eclipse y desarrollado por Aptana, Inc., que puede funcionar bajo Windows, Mac y Linux y provee soporte para lenguajes como: Php, Python, Ruby, CSS, Ajax, HTML y Adobe AIR. Tiene la posibilidad de incluir complementos para nuevos lenguajes y funcionalidades.

Algunas de sus principales características son las siguientes:

- Asistente de código para HTML y Javascript.
- Librerías ajax (jQuery, prototype, scriptaculous, Ext JS, dojo, YUI y Spry entre otras).
- Conexión vía FTP, SFTP, FTPS y Aptana Cloud.
- Herramientas para trabajo con base de datos.
- Marcado de sintaxis mediante colores.
- Compatible con extensiones para Eclipse (existen más de 1000).

Sitio oficial: www.aptana.com

5.2 **Tortoise SVN**

TortoiseSVN es un software para desarrolladores liberado bajo la licencia GNU (General PublicLicence). Es utilizado por estos para gestionar versiones del código fuente de sus programas.

TortoiseSVN es un cliente subversión implementado como una extensión de la línea de comandos de Microsoft por lo que resulta muy fácil de utilizar.

Este software gratuito nos permite trabajar con un repositorio remoto común en el cual podemos realizar el control de versiones de nuestro sistema.

Algunas de sus principales características son:

- Integración con la línea de comandos de Windows.
- Puede ser usado sin un entorno de desarrollo.
- Pequeñas imágenes decoran los íconos de los archivos mostrando qué archivos o directorios necesitan ser enviados al repositorio.
- Disponible en 28 idiomas diferentes.
- Maneja el mostrar la diferencia de documentos de Office tales como los creados con Microsoft Word.
- Fácil acceso a los comandos de Subversión: Todos los comandos de Subversión están disponibles desde el menú contextual del explorador. TortoiseSVN añade su propio submenú allí.
- Versionado de carpetas

- Confirmaciones atómicas: Una confirmación puede entrar al repositorio completamente, o no entrar en absoluto. Esto permite a los desarrolladores generar y confirmar cambios como unidades lógicas.
- Metadatos versionados
- Cada archivo y carpeta tiene un conjunto invisible de “propiedades” adjuntos. Se puede crear y almacenar cualquier par de clave/valor que se desee. Las propiedades se versionan en el tiempo, igual que el contenido de los archivos.
- Elección de capas de red: Subversión tiene una noción abstracta del acceso al repositorio, haciendo que se puedan implementar nuevos mecanismos de red fácilmente.
- Manejo de datos consistente.
- Etiquetado y creación de ramas eficiente

Sitio Oficial: <http://tortoisesvn.net/>

5.3 WAMP Server

WAMP Server es un entorno de desarrollo para Windows con el que se puede crear aplicaciones web con Apache, PHP y bases de datos MySQLdatabases. También incluye PHPMyAdmin y SQLiteManager para manejar bases de datos en forma sencilla y eficaz.

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo;
- Apache, como servidor web;
- MySQL, como gestor de bases de datos;
- PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un WAMP permite servir páginas html a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web.

Sitio oficial: <http://www.wampserver.es/>

5.4 MySQLWorkbench

MySQLWorkbench es una herramienta visual unificada para los arquitectos de bases de datos, desarrolladores y administradores de bases. MySQLWorkbench ofrece modelado de datos, desarrollo de SQL y herramientas completas de administración para la configuración del servidor, la administración de usuarios, copia de seguridad, y mucho más. MySQLWorkbench está disponible en Windows, Linux y Mac OS X.

MySQLWorkbench permite a un DBA, desarrollador o arquitecto de datos diseñar visualmente, modelar, generar y gestionar bases de datos. Incluye todo lo que un modelador de datos necesita para la creación de modelos de ER complejos, ingeniería directa e inversa, y también

ofrece funciones clave para la realización de las tareas más difíciles de gestión del cambio y la documentación que normalmente requieren mucho tiempo y esfuerzo.

MySQLWorkbench proporciona herramientas visuales para crear, ejecutar, y optimizar consultas SQL. El Editor SQL proporciona un color resaltado de sintaxis, auto-completado, la reutilización de fragmentos de código SQL, y la historia de ejecución de SQL. El panel de conexiones de base de datos permite a los desarrolladores para gestionar fácilmente las conexiones de base de datos. El Examinador de objetos proporciona acceso instantáneo a esquema y objetos de base de datos.

Sitio Oficial:<http://www.mysql.com/products/workbench/>

5.5 Cliente de base de datos. Navicat.

NavicatforMySQL es una solución ideal para la administración y el desarrollo de MySQL/ MariaDB. Permite la conexión a bases de datos MySQL y MariaDB simultáneamente desde una sola aplicación. Este producto ofrece una intuitiva y potente interfaz gráfica de gran alcance para la gestión, el desarrollo y el mantenimiento de las bases de datos.

NavicatforMySQL se conecta a cualquier servidor MySQL o MariaDB ya sea local o remoto. Funciona con cualquier servidor de base de datos MySQL, desde la versión 3.21 o superior y MariaDB 5.1 o superior. También es compatible con Drizzle, OurDelta y Percona Server. También es compatible con la mayoría de las últimas características incluyendo tablas, vistas, funciones / procedimientos, eventos, y mucho más.

Las características principales incluyen Generador / Editor de SQL, herramienta de Modelado de Datos, Transferencia de datos, Importación / Exportación, Sincronización de Datos y Estructura, Informes y mucho más.

Sitio oficial: <http://www.navicat.com/es/>

5.6 Herramientas de Firefox. Firebug.

Firebug es una extensión de Firefox creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura DOM), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea e inline.

Firebug no es un simple inspector como DOM Inspector, además edita y permite guardar los cambios, un paso por delante del conocido Web Developer. Su atractiva e intuitiva interfaz, con solapas específicas para el análisis de cada tipo de elemento (consola, HTML, CSS, Script, DOM y red), permite al usuario un manejo fácil y rápido. Firebug está encapsulado en forma de plug-in o complemento de Mozilla, es Open Source, libre y de distribución gratuita.

Sitio oficial: <http://getfirebug.com/>

6 SprintBacklog

6.1 Sprint 1

En este primer sprint comenzamos con la idea de definir las métricas necesarias para seleccionar la cantidad de historias a incluir. Como nos encontramos dentro de la primera iteración no contamos con el tiempo utilizado en el sprint anterior para calcular el tiempo de dedicación, por lo cual definimos que el mismo sería de 15 días (sin incluir sábados, domingos y días no laborables), y un total de 24 SP.

Period: may 6 - may 24 (15 days)

Product Owner: bcitroni, mpierini

Scrum Master: bcitroni, mpierini

Team: bcitroni, mpierini

Metas del Sprint 1

- Establecer las pantallas principales del Sistema
- Definir módulos con gran caudal de información.
- Establecer el modelo de datos inicial
- Establecer el plan de pruebas del proyecto

Historia	Prioridad	Estimación
0 Establecer el ProductBacklog Inicial del Sistema	1	4
1 Diseño de interfaces principales	1	7
2 Creación Del Modelo de Datos	1	2
3 Gestionar Asistencias	2	6
4 Gestionar Alumnos	2	4
5 Creación Plan de Pruebas	3	1
Total Sprint 1		24

Como se realiza en todos los Sprint, en primer lugar comenzamos con una reunión con el propietario del producto, en la cual se seleccionaron las historias prioritarias a incluir en esta primera iteración.

Luego de finalizada la reunión, definimos las prioridades de estas historias.

Tarea	Responsable	Estimación	Real
HISTORIA 0– Establecer el ProductBacklog Inicial del Sistema			
✓ Establecer las historias de usuario iniciales.	BC, MP	2	3
✓ Estimación de historias.	BC, MP	2	2

Tarea	Responsable	Estimación	Real
HISTORIA 1 – Diseño de interfaces principales			
✓ Diseñar la estructura principal.	BC, MP	1	2
✓ Diseño pantalla "home".	BC, MP	3	4
✓ Diseñar boceto de pantallas varias.	BC, MP	2	2
✓ Validar pantallas con el cliente	BC, MP	1	1

Para todo lo que está relacionado con las interfaces decidimos implementar un formato de diseño que actualmente está siendo muy utilizado dentro del mundo del desarrollo web. El mismo es conocido como "diseño responsivo" y nos da la posibilidad de utilizar el sistema en distintos dispositivos manteniendo la visibilidad de los componentes del mismo. Para más detalle sobre esto se podrá consultar el anexo número 1 –"Diseño Web Responsivo o Responsive Web Design."-.

Tarea	Responsable	Estimación	Real
HISTORIA 2 – Creación Del Modelo de Datos			
✓ Definir tablas y sus relaciones.	BC, MP	1	2
✓ Depurar modelo de datos.	BC, MP	1	1

Tarea	Responsable	Estimación	Real
HISTORIA 3 – Gestionar Asistencias			
✓ Definir actividades sobre el modulo	MP	1	2
✓ Definir validaciones	MP	1	1
✓ Desarrollo de las interacciones del lado del cliente	MP	2	2
✓ Desarrollo de la lógica de la aplicación del lado del servidor	MP	2	2

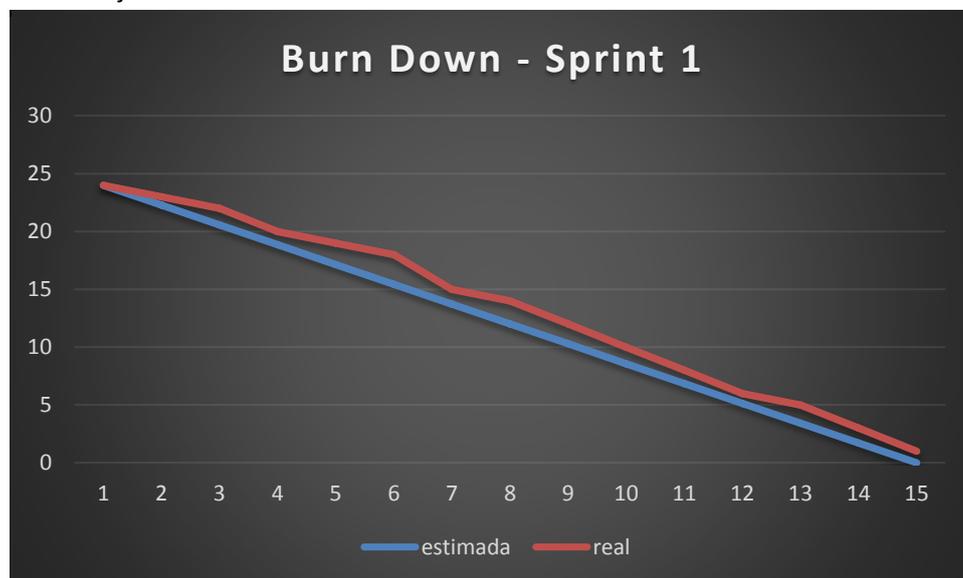
Tarea	Responsable	Estimación	Real
HISTORIA 4 – Gestionar Alumnos			
✓ Definir actividades sobre el modulo	BC	0.5	0.5
✓ Definir validaciones	BC	0.5	0.5
✓ Desarrollo de las interacciones del lado del cliente	BC	1.5	2
✓ Desarrollo de la lógica de la aplicación del lado del servidor	BC	1	2
✓ Revisión de las interfaces. Correcciones Mínimas	BC	0.5	0.5

Tarea	Responsable	Estimación	Real
HISTORIA 5– Creación Plan de Pruebas			
✓ Crear plan de pruebas general.	BC, MP	1	1

Este primer sprint fue muy productivo ya que nos permitió, como equipo de trabajo, poder focalizarnos en varios aspectos que nos iban a servir como base para los desarrollos posteriores. La participación del cliente nos sirvió para ir resolviendo cualquier incógnita que se nos presentó a medida que íbamos avanzando en el sprint. También sirvió para entrar en contacto con la metodología en un ambiente real de desarrollo y de esta forma comenzar a aprovechar sus ventajas para tener un proceso de desarrollo mucho más eficiente.

A continuación mostraremos el “Burn Down” del primer sprint.

Dicha gráfica es obtenida de la comparación de los tiempos estimados o planificados contra los tiempos de trabajo real.



6.2 Sprint 2

Period: may 27 - jun14 (15 days)
 Product Owner: bcitroni, mpierini
 Scrum Master: bcitroni, mpierini
 Team: bcitroni, mpierini

Metas del Sprint 2

- Tener las interfaces principales desarrolladas
- Desarrollo del AB de Usuarios.
- Definición de mecanismos de seguridad.

	Historia	Prioridad	Estimación
6	Diseño de interfaces del sistema	3	12
7	Diseño del menú principal	3	2
8	Diseñar interfaz de login	3	1
9	Definir permisos de usuarios	4	1
10	Gestionar Usuarios	5	5
11	Definir políticas de seguridad	5	1
Total Sprint 2			22

Tarea	Responsable	Estimación	Real
HISTORIA 6 – Diseño de interfaces del sistema			
✓ Desarrollo de las ventanas de la aplicación.	BC, MP	5	6
✓ Diseño grafico	BC, MP	3	3
✓ Desarrollo de la navegación del menú a las distintas interfaces	BC, MP	4	4

Tarea	Responsable	Estimación	Real
HISTORIA 7 – Diseño del menú principal			
✓ Diseño de la interface del menú principal	MP	1	2
✓ Diseño grafico	MP	1	1

Tarea	Responsable	Estimación	Real
HISTORIA 8 – Diseñar interfaz de login			
✓ Diseño de la interface de login	BC	1	1
✓ Diseño grafico	BC	1	0.5

Descripción de la Historia 8:

Pensamos la mejor forma de desarrollar una interfaz de login que sea sencilla pero al mismo tiempo tenga todo lo necesario para poder cumplimentar con los requisitos propios de una aplicación de este tipo. Mediante dicha interfaz se podrá acceder al sistema con un usuario y contraseña, que deberá ser creado con anterioridad. Dicho perfil al mismo tiempo tendrá asignado diferentes permisos que habilitaran al usuario a utilizar distintas funcionalidades del aplicativo.

Datos a solicitar por el sistema:

- Usuario
- Contraseña

Tarea	Responsable	Estimación	Real
HISTORIA 9 – Definir permisos de usuarios			
✓ Definir perfiles de usuarios	MP	1	1

Tarea	Responsable	Estimación	Real
HISTORIA 10 – Gestionar Usuarios			
✓ Definir actividades sobre el modulo	BC	0.5	0.5
✓ Definir validaciones	BC	1	1
✓ Desarrollo de las interacciones del lado del cliente	BC	1.5	2
✓ Desarrollo de la lógica de la aplicación del lado del servidor	BC	1.5	2
✓ Revisión de las interfaces. Correcciones Mínimas	BC	0.5	0.5

Tarea	Responsable	Estimación	Real
HISTORIA 11 – Definir políticas de seguridad			
✓ Diseñar mecanismo de seguridad	BC	0.5	1
✓ Definir perfiles de seguridad en base a perfiles	BC	0.5	0.5

Notas Retrospectivas:

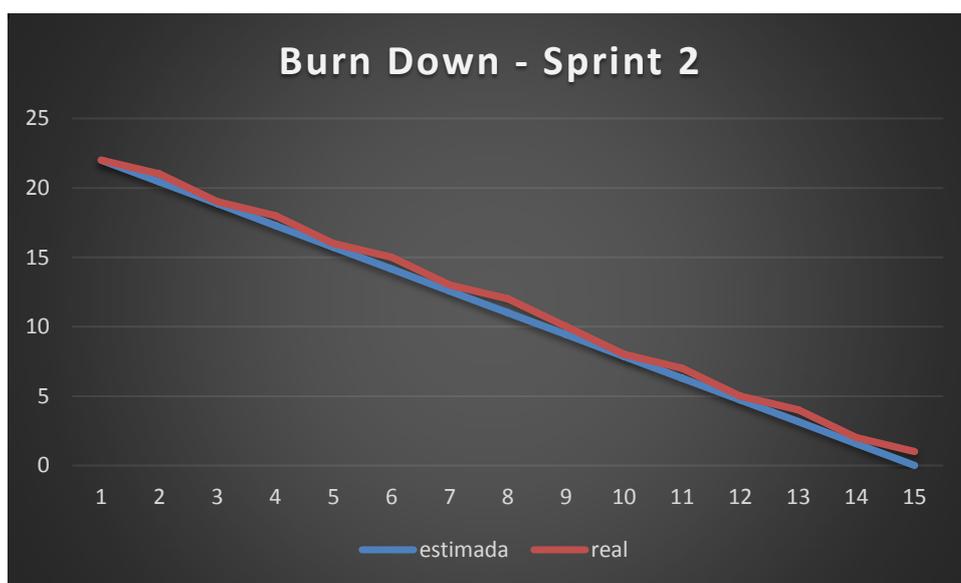
Lo bueno

- Familiarización con herramientas de diseño
- Mayor orden en el trabajo
- Hubo un aprendizaje de plantillas de estilos y de mecanismos de seguridad

A mejorar

- Organización y planificación del trabajo y división de tareas.
- Se debe, en lo posible, realizar un commit diario del código, siempre que compile sin errores.

Gráfica o “Burn Down” con la relación entre lo estimado y el trabajo real:



6.3 Sprint 3

Period: jun 17 - jul5 (15 days)

Product Owner: bcitroni, mpierini

Scrum Master: bcitroni, mpierini

Team: bcitroni, mpierini

Metas del Sprint 3

- Tener los siguientes ABM terminados de:
 - ✓ Barrios
 - ✓ Grupos
 - ✓ Centro de educación.
- Comenzar con la población de la base de datos con información proveniente del movimiento.

	Historia	Prioridad	Estimación
12	Gestionar Barrios	6	7
13	Gestionar Centro de Educación	6	3
14	Gestionar Grupos	6	5
15	Poblar BD con datos	8	4
Total Sprint 3			19

Tarea	Responsable	Estimación	Real
HISTORIA 12 – Gestionar Barrios			
✓ Definir actividades sobre el modulo	BC	1	1.5
✓ Definir validaciones	BC	1	1
✓ Desarrollo de las interacciones del lado del cliente	BC	2	2.5
✓ Desarrollo de la lógica de la aplicación del lado del servidor	BC	2	2.5
✓ Revisión de las interfaces. Correcciones Mínimas	BC	1	1

Descripción de la Historia **Gestionar Barrios**.

Como administrador deseo registrar / modificar un barrio dentro del sistema de educación.

Dentro de esta historia deseo poder nos solamente poder crear un nuevo barrio sino que también poder crear y agregar distintas propiedades al barrio que luego puedan ser motivo de consulta y evaluación.

El poder contar con la descripción de los distintos barrios me dará la posibilidad de tomar distintas decisiones que puedan influir directa o indirectamente en la asignación de recursos para realizar mejoras en el barrio que presente mayores necesidades.

Test de Aceptación:

El barrio no debe existir en el modelo de datos.

Tarea	Responsable	Estimación	Real
HISTORIA 13 – Gestionar centros de educación			
✓ Definir actividades sobre el modulo	MP	0.5	0.5
✓ Definir validaciones	MP	0.25	0.5
✓ Desarrollo de las interacciones del lado del cliente	MP	1	1.5
✓ Desarrollo de la lógica de la aplicación del lado del servidor	MP	1	1
✓ Revisión de las interfaces. Correcciones Mínimas	MP	0.25	0.5

Tarea	Responsable	Estimación	Real
HISTORIA 14 – Gestionar grupos			
✓ Definir actividades sobre el modulo	MP	1	1
✓ Definir validaciones	MP	0.5	0.5
✓ Desarrollo de las interacciones del lado del cliente	MP	1.5	2
✓ Desarrollo de la lógica de la aplicación del lado del servidor	MP	1.5	2
✓ Revisión de las interfaces. Correcciones Mínimas	MP	0.5	1

Tarea	Responsable	Estimación	Real
HISTORIA 15 – Poblar BD con datos			
✓ Definir información para migrar	BC, MP	1	1
✓ Definir metodología de migración	BC, MP	1	1
✓ Analizar información a migrar	BC, MP	2	2.5

Notas Retrospectivas:

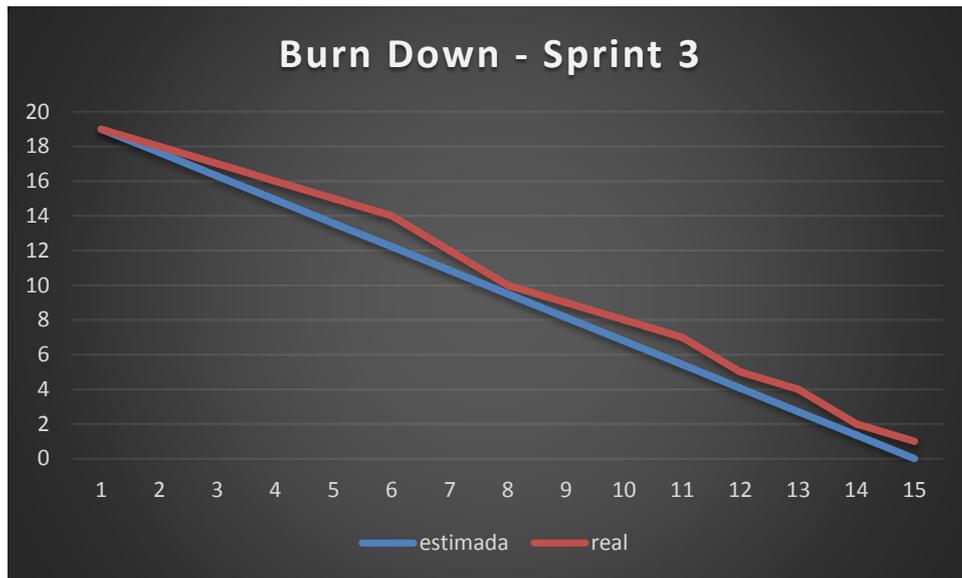
Lo bueno

- El equipo se siente más seguro y confiado con la tecnología y puede avanzar a mejor ritmo del que lo venía haciendo con anterioridad.
- Visibilidad del avance del proyecto más clara.

A mejorar

- Hubo varios cambios propuestos por el cliente.

Gráfica generada con la relación entre lo estimado y el trabajo real:



6.4 Sprint 4

Period: jul 8 - jul 26 (15 days)
 Product Owner: bcitroni, mpierini
 Scrum Master: bcitroni, mpierini
 Team: bcitroni, mpierini

Metas del Sprint 4

- Tener los siguientes módulos terminados de:
 - ✓ Evaluaciones
 - ✓ Avisos
 - ✓ Estadísticas (sobre asistencias y evaluaciones)
- Comenzar a definir las representaciones graficas asociadas a la información contenida en el sistema.

HISTORIA	Prioridad	Estimación
16 Gestionar evaluaciones	6	6
17 Gestionar estadísticas asistencias	8	7
18 Gestionar estadísticas evaluaciones	8	7
19 Gestionar avisos	9	2
20 Definir representaciones graficas sobre estadísticas	9	4
		26

Tarea	Responsable	Estimación	Real
HISTORIA 16 – Gestionar evaluaciones			
✓ Definir actividades sobre el modulo	BC	1	1
✓ Definir validaciones	BC	0.5	1
✓ Desarrollo de las interacciones del lado del cliente	BC	2	2..5
✓ Desarrollo de la lógica de la aplicación del lado del servidor	BC	2	3
✓ Revisión de las interfaces. Correcciones Mínimas	BC	0.5	0.5

Tarea	Responsable	Estimación	Real
HISTORIA 17 – Gestionar estadísticas asistencias			
✓ Definir actividades sobre el modulo	MP	1	1
✓ Definir validaciones	MP	0.5	1
✓ Desarrollo de las interacciones del lado del cliente	MP	2	3
✓ Desarrollo de la lógica de la aplicación del lado del servidor	MP	2	3
✓ Revisión de las interfaces. Correcciones Mínimas	MP	0.5	1

Tarea	Responsable	Estimación	Real
HISTORIA 18 – Gestionar estadísticas evaluaciones			
✓ Definir actividades sobre el modulo	BC	1	1
✓ Definir validaciones	BC	0.5	1
✓ Desarrollo de las interacciones del lado del cliente	BC	2	3
✓ Desarrollo de la lógica de la aplicación del lado del servidor	BC	2	3
✓ Revisión de las interfaces. Correcciones Mínimas	BC	0.5	1

Descripción de las Historias Gestionar estadísticas asistencias y Gestionar estadísticas evaluaciones:

Ambas historias tienen un papel preponderante en el sistema ya que no solamente permiten reflejar toda la información que sea cargada en el mismo sino que también darán la posibilidad, al administrador, de poder tomar decisiones de vital importancia para la vida del movimiento. Esto nos permitirá mejorar la calidad de aprendizaje de todos los alumnos que participen de las distintas opciones educativas que ofrece el movimiento en la actualidad.

Dentro de ambas historias se deberán tener las siguientes opciones:

- Combo que permitan seleccionar las estadísticas a generar.

- Representar las estadísticas mediante graficas que sean sumamente fáciles de comprender
- Exportar las gráficas y estadísticas en reportes capaces de ser utilizados según sea necesario.

Tarea	Responsable	Estimación	Real
HISTORIA 19 – Gestionar avisos			
✓ Definir actividades sobre el modulo	MP	1	1
✓ Definir validaciones	MP	0.5	0.5
✓ Desarrollo de las interacciones del lado del cliente	MP	2	2
✓ Desarrollo de la lógica de la aplicación del lado del servidor	MP	2	2.5
✓ Revisión de las interfaces. Correcciones Mínimas	MP	0.5	0.5

Tarea	Responsable	Estimación	Real
HISTORIA 20 – Definir representaciones graficas sobre estadísticas			
✓ Definir gráficas a implementar	BC, MP	0.5	0.5
✓ Analizar herramientas de generación de gráficas	BC, MP	1	2
✓ Definir información asociada a cada gráfica	BC, MP	1	2
✓ Analizar requerimientos del cliente	BC, MP	0.5	0.5

Notas Retrospectivas:

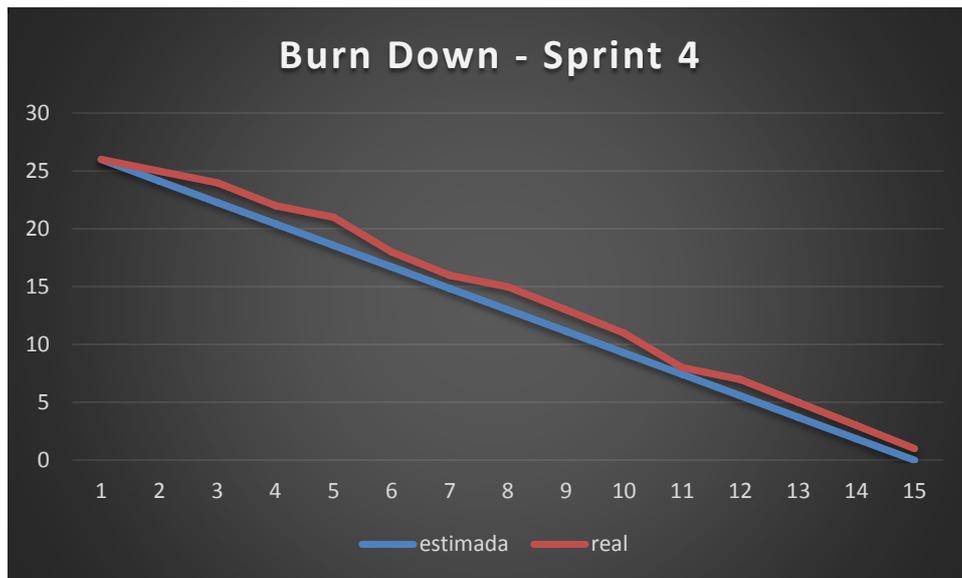
Lo bueno

- Aprendizaje de herramientas capaces de generar gráficos que representen la información contenida en el sistema.
- Utilización de métodos para generar reportes con las estadísticas obtenidas en las distintas historias.
- Se comenzaron a realizar commit en forma más frecuente, lo cual permitió tener un mayor control sobre el código generado.

A mejorar

- Organización de los tiempos de trabajo y alcance de algunas tareas.

Gráfica generada con la relación entre lo estimado y el trabajo real:



6.5 Sprint 5

Period: jul 29 - agost 16 (15 days)

Product Owner: bcitroni, mpierini

Scrum Master: bcitroni, mpierini

Team: bcitroni, mpierini

Metas del Sprint 5

- Tener el sistema probado y funcionando correctamente
- Refinar las interfaces desarrolladas.
- Generar reportes capaces de reflejar la información más importante.

Historia	Prioridad	Estimación
21 Cerrar listado de Requerimientos	9	1
22 Crear Reportes	9	5
23 Diseñar interfaces adaptables	10	5
24 Refinar Diseño de Interfaces	11	2
25 Probar el Sistema	11	5
26 Aplicar políticas de mejoras de rendimiento	12	2
Total Sprint 5		20

Tarea	Responsable	Estimación	Real
HISTORIA 21 – Cerrar listado de requerimientos			
✓ Definir requerimientos de último momento	BC, MP	1	1

Descripción de la historia Cerrar listado de requerimientos:

- Definir y documentar las Historias de Usuario faltantes, y priorizarlas con el Scrum Master
- Revisar historias de usuario actuales y ver lo que sobra o falta, completando las historias faltantes y borrando las que ya no son necesarias.

Tarea	Responsable	Estimación	Real
HISTORIA 22 – Crear Reportes			
✓ Diseñar de reportes en formato requerido	BC	2	2
✓ Definir herramienta de generación de reportes	BC	2	2.5
✓ Analizar información a incluir en cada reporte	BC	1	1

Tarea	Responsable	Estimación	Real
HISTORIA 23 – Diseñar interfaces adaptables			
✓ Diseñar interfaces responsivas	BC	2	2
✓ Desarrollo de las interfaces	BC	3	5

Tarea	Responsable	Estimación	Real
HISTORIA 24 – Refinar diseño de interfaces			
✓ Analizar posibles mejoras	MP	1	1
✓ Revisar interacción entre interfaces	MP	1	1.5

Tarea	Responsable	Estimación	Real
HISTORIA 25 – Probar el sistema			
✓ Realizar las pruebas funcionales correspondientes	BC, MP	2	3
✓ Generar reporte	BC, MP	1	1
✓ Validar resultados con el cliente	BC, MP	2	4

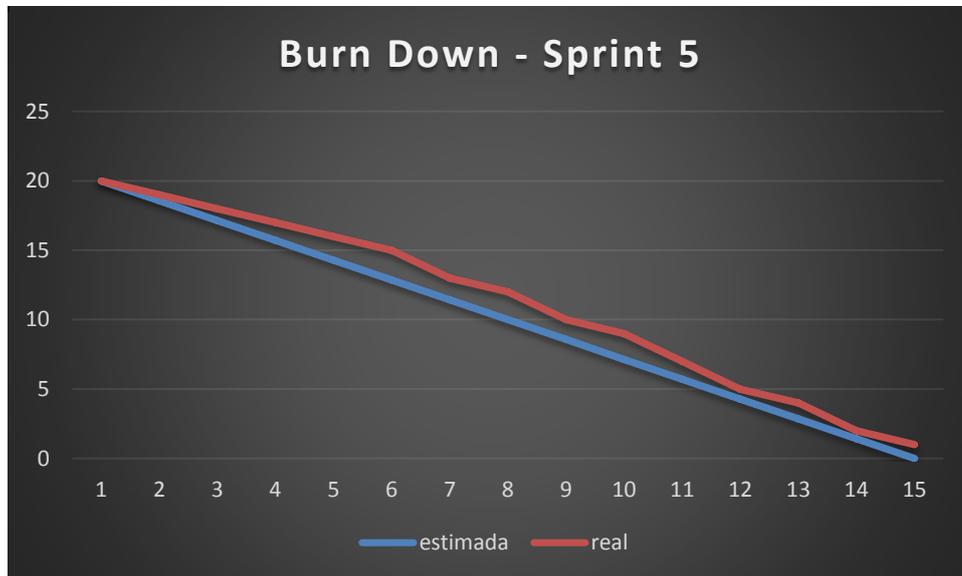
Descripción de la tarea Validar resultados con el cliente:

Mediante esta tarea se procederá en conjunto con el cliente a un análisis formal de lo desarrollado y en base a esto se podrá medir el nivel de aceptación del cliente con respecto al proyecto en general.

Tarea	Responsable	Estimación	Real
HISTORIA 26 – Aplicar políticas de mejora de rendimiento			

✓ Analizar posibles mejorar de funcionalidades	BC, MP	1	1
✓ Aplicar mecanismos de performance	BC, MP	1	2

Gráfica generada con la relación entre lo estimado y el trabajo real:



7 Migración de Datos

Esta etapa del proyecto toma un papel preponderante sobre todo cuando se trata de un sistema que se nutre constantemente de toda la información que se puede ir recopilando día a día en cada centro educativo. Todo esto se puede ir reflejando en cada una de las estadísticas que se pueden obtener en cada uno de los módulos que componen el sistema de educación.

Hoy en día muchos de los datos con los cuales se maneja la organización están en papel, y eso es lo que hace imposible su aprovechamiento. Es por esto que la planificación de una migración de datos de este tipo es más compleja que cualquier otra. Aquí se debe plantear una migración mucho más exhaustiva, ya que el trabajo que se debe realizar es mucho más "pesado", y toda la información que se debe cargar en la base de datos proviene de papeles.

Del análisis realizado vimos que la migración sería un proceso largo, en el cual no solamente estaríamos involucrados nosotros, sino que también la gente del movimiento debería participar en la misma. Dicha participación les serviría como método de capacitación, mediante el cual podrían ir conociendo cada uno de los módulos del sistema e ir realizando la carga de los datos que ellos mismos capturan todos los días.

Como consecuencia de todo lo mencionado anteriormente planificamos una migración de datos que sea capaz de privilegiar las necesidades de información más importantes del movimiento, para de esta forma poder comenzar a explotar las potencialidades del sistema de educación en

forma rápida y eficaz. Se comenzaría con la migración de aquellos datos que, luego de un proceso de determinación por parte del movimiento, puedan brindar información relevante sobre la actual situación del sector educativo. Como primera medida decidimos la migración de todas las planillas de asistencia, de cada grupo educativo, para de esta forma poder empezar a realizar un control sobre la asistencia de los alumnos a las distintas clases que ofrece el movimiento.

El proceso de migración fue pensado como un proceso incremental en el cual a medida que se vayan estabilizando los diferentes módulos, se irá incorporando más información en los mismos. A medida que la migración de toda la información que hoy se tiene en planillas se fuera completando, el sistema le ira permitiendo a los directores del movimiento valerse del mismo para recopilar información que sea importante al momento de tomar decisiones, no solo a corto plazo, sino que también se permitirá armar un plan a mediano y largo plazo.

8 Pruebas de software

8.1 **Scrum y Testing**

Durante los sprints la aplicación es diseñada, codificada y **testeada**.

Los testers forman parte del ScrumTeam: En cada sprint se desarrolla y se prueba la funcionalidad establecida para cumplir el sprint goal, por lo cual, existirá un grupo de testers dentro del equipo que prepare, especifique y ejecute los tests necesarios. Testers y desarrolladores pueden intercambiar roles entre sprints, e incluso durante el sprint en curso con el objetivo fundamental de cumplir con el sprint goal; claro está que esta práctica no se alinea a las mejores prácticas de testing.

Es fundamental evaluar la conformidad funcional del producto de software, es decir, evaluar su correcto funcionamiento de acuerdo con aquellas especificaciones que fueron establecidas para la misma, y es por ello que se diseñan las pruebas funcionales pertinentes.

8.1.1 **Rol del Tester**

8.1.1.1 Sprint Planning Meeting

Durante esta reunión el Tester realiza preguntas y comentarios sobre los ítems del SprintBacklog que serán incluidos en el sprint para definir el alcance de la funcionalidad a ser probada y dar una primera estimación de estas pruebas. Luego deberá definir las tareas de testing que serán incluidas en el SprintBacklog.

8.1.1.2 Durante el Sprint

Provee la conciencia de calidad en el equipo. Es responsable de comunicar los riesgos que se van detectando durante las pruebas para medir junto al resto del equipo la calidad del producto que se está construyendo.

8.1.1.3 Review Meeting

Da visibilidad al equipo sobre los posibles resultados de esta reunión y define el “camino feliz” que se utilizará durante la presentación, que usualmente conduce.

8.1.1.4 Retrospective Meeting

Aporta la verdadera visión sobre la calidad del producto entregado y el proceso de desarrollo en general, y presenta sus conclusiones sobre el proceso de pruebas, indicando posibles mejoras al mismo.

8.2 Pruebas aplicadas al sistema.

Una vez analizado todo aquello que está relacionado con la ejecución de pruebas en el marco teórico que nos brinda Scrum, procederemos a explicar acerca del paquete de test aplicado a nuestro sistema de información, y más en detalle se explicarán los distintos tipos de pruebas que lo compusieron.

Vale destacar que las tareas relacionadas con el testing de la aplicación fueron realizadas por nosotros mismos, quienes en conjunto y con la participación especial del cliente en las distintas etapas que las conforman, hemos analizado el resultado de las mismas, pero por sobre todo realizando una verificación de los resultados que estas iban arrojando, para de esta forma ir logrando, a medida que se sucedieran los sprints, una aplicación que realmente cumpla con las expectativas del movimiento Los Sin Techo.

8.2.1 Pruebas de unidad.

Las pruebas unitarias son una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las pruebas de integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

Para que una prueba unitaria sea buena se deben cumplir los siguientes requisitos:

- Automatizable.
- No debería requerirse una intervención manual, lo cual es especialmente útil para integración continua.
- Completas.
- Deben cubrir la mayor cantidad de código.
- Repetibles o Reutilizables.
- No se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para integración continua.
- Independientes: la ejecución de una prueba no debe afectar a la ejecución de otra.
- Profesionales

El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el trozo de código debe satisfacer.

Las pruebas aisladas proporcionan cinco ventajas básicas:

- **Fomentan el cambio**
- **Simplifica la integración**
- **Documenta el código**
- **Separación de la interfaz y la implementación**

➤ **Los errores están más acotados y son más fáciles de localizar**

A pesar de dichas ventajas, es importante darse cuenta de que las pruebas unitarias no descubrirán todos los errores del código. Por definición, sólo prueban las unidades por sí solas. Por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto.

Para nuestro aplicativo dichas pruebas las realizamos bajo nuestra órbita, lo cual nos permitió ir creando diferentes casos de test que se enfocaban en las unidades de software, que normalmente son los métodos utilizados durante el desarrollo del sistema. Dichos métodos eran testeados en forma aislada, es decir, sin verificar la interacción de los mismos con diferentes componentes.

Estos test nos permitían focalizar nuestros esfuerzos en lograr componentes que realicen su trabajo de la forma más eficiente y eficaz posible.

8.2.2 Pruebas de integración.

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de verificar que interactúan correctamente a través de sus interfaces, tanto internas como externas, que cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

Los tipos fundamentales de integración son los siguientes:

Integración incremental: se combina el siguiente componente que se debe probar con el conjunto de componentes que ya están probados y se va incrementando progresivamente el número de componentes a probar.

Integración no incremental: se prueba cada componente por separado y posteriormente se integran todos de una vez realizando las pruebas pertinentes.

En lo que respecta a este tipo de pruebas, las mismas fueron realizadas sobre los distintos módulos que componen el sistema de educación. Al utilizar una metodología de desarrollo como Scrum, esta nos asegura la posibilidad de poder ir realizando el testing sobre cada uno de los sprints que vayan finalizando. De esta forma podíamos acotar cualquier error encontrado a la funcionalidad agregada en dicho Sprint y por sobre todo podíamos obtener rápida realimentación sobre el resultado final del sprint y lo que este nos proporcionó como módulo funcional del aplicativo.

Además, esta estrategia nos dio la posibilidad de comenzar cada nuevo Sprint con un cierto nivel de seguridad y confianza sabiendo que partíamos de un producto que funcionaba correctamente.

8.2.3 Pruebas funcionales.

Las pruebas funcionales son aquellas que se ocupan de probar y validar que el sistema hace lo que debe y sobre todo, lo que se ha especificado. Son pruebas específicas, concretas y exhaustivas realizadas por el cliente.

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que el objetivo no es centrar la atención en cómo se generan las respuestas del sistema, sino que el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida.

Las pruebas funcionales se realizaron al finalizar cada Sprint, previa a la reunión de retrospectiva. Estas pruebas se realizaron de forma manual, y se hicieron sobre los procesos importantes o mejor dicho, sobre aquellos procesos que revestían calidad de “primordial” para el movimiento Los Sin Techo.

Las reuniones de retrospectiva, mencionadas anteriormente, contaban con un alto grado de importancia, ya que utilizábamos a las mismas como para cerrar un ciclo y de esta forma poder comenzar uno nuevo. En aquellos Sprint en los cuales no se aplicaban cambios de funcionalidad se intentaba realizar pruebas de aceptación de usuario para poder cerrar el ciclo de producción del Sprint y continuar con los restantes. Todo lo anteriormente descrito fue mencionado como una intención, ya que muchas veces el cliente, al no contar con gran cantidad de recursos (humanos, tiempo y espacio, etc.) no disponía de los tiempos necesarios para realizar dichas pruebas.

9 Plan de Implementación

Para la implementación del sistema de información decidimos utilizar una metodología de trabajo que consistía en primera instancia de un proceso incremental en el cual a medida que se iba finalizando cada uno de los sprints, dicho incremento era revisado con el cliente, para de esta forma obtener el feedback final sobre el mismo.

Al momento de finalizar cada uno de los sprints se procedía a realizar, en conjunto con el cliente, una serie de pruebas que nos permitían conocer el estado de satisfacción del cliente a medida que avanzaba el proyecto. Cada una de estas instancias tiene como resultado un feedback que se traducía en mejoras y correcciones del sistema que se veían reflejadas en el siguiente sprint.

Desde un primer momento se acordó con la gente del movimiento, que más allá de ir realizando entregas parciales del producto para poder obtener un feedback asociado a las mismas, el “golive” del sistema de educación se haría una vez que el mismo esté completado en su totalidad. Uno de los principales motivos de esta decisión fue el hecho de que el movimiento había realizado hace relativamente muy poco tiempo una mudanza de sus instalaciones, en cuanto a tecnología, lo cual hacía bastante difícil el poder contar con los recursos necesarios para ir realizando la puesta en producción de cada sprint una vez que haya sido finalizado.

Para poder alcanzar la implementación total del sistema de educación se planificaron distintas etapas que sirvieron como método de guía para alcanzar con éxito el objetivo que se había acordado con la gente del movimiento. Dichas etapas permitieron que la implementación del sistema de información puede ser realizada en forma ordenada y eficaz.

Una vez finalizado el último sprint y las pruebas correspondientes, procedimos a realizar la implementación total del sistema en los equipos del cliente.

10 Conclusión

Desde todo punto de vista este proyecto se presentó desde un principio como un verdadero desafío, no solo por todo aquello que engloba un verdadero sistema de información, sino que también se nos presentó como desafío el representar, en mayor o menor medida, el rol que deben cumplir los profesionales dentro de la sociedad donde desarrollan su actividad. Desde un primer momento vimos la posibilidad de trabajar en conjunto con el movimiento “Los Sin Techo” como la oportunidad para poder aportar, desde nuestra perspectiva, un granito de arena que ayude a colaborar, en este caso en particular, con una ONG que tiene como principal iniciativa la inclusión social.

Si ahora vemos el proyecto desde una órbita totalmente técnica/profesional todo lo que hemos realizado durante el periodo que duró este proyecto nos sirvió no solo para poder afianzar conocimientos que ya poseíamos, sino que también nos permitió acercarnos a nuevos conocimientos y de esta forma poder incorporarlos a nuestra formación académica y laboral.

Este crecimiento también se vio favorecido con la posibilidad de trabajar en conjunto con el cliente, en este caso el movimiento “Los Sin Techo”. Esta experiencia fue algo que nos permitió aprender a interactuar con un cliente real, más allá de alguna que otra experiencia laboral con la que contábamos. Poder comprender lo que el cliente quería fue vital, esto nos sirvió para poder entender lo que el cliente realmente necesitaba, y en base a esto tratar de ofrecerle una solución integral que le permita al movimiento dar un paso hacia adelante en cuanto a la “informatización” de toda la información con la que cuenta.

Una de las principales novedades que pudimos aplicar en este nuevo proyecto que emprendimos fue la implementación de las metodologías ágiles. En este caso en particular decidimos utilizar Scrum, y nos pusimos como objetivo el poder explotar todos beneficios que posee, para lograr un sistema de información que cuente con altos índices de calidad, pero por sobre todo que cumpla con los requisitos del cliente. La organización del trabajo en sprints resultó un factor clave para llevar adelante el desarrollo del sistema, puesto que con cada incremento implementado obteníamos un feedback importantísimo que nos permitió no solo tener la chance de anticiparnos a problemas futuros sino que también nos ayudó a ver el grado de satisfacción que el cliente iba teniendo a medida que veía dichos incrementos. La base sólida para poder enfocarnos en los requerimientos fueron las reuniones que fuimos manteniendo con el movimiento, las cuales nos permitieron ir limando aquellas dudas o dificultades que iban surgiendo a medida que avanzábamos con el desarrollo.

Desde que iniciamos este proyecto nos pusimos como objetivo, no solo poder concluir nuestra formación académica y todo lo que ello conlleva, sino que también nos enfocamos en tratar de ayudar a la ONG desde nuestra perspectiva de futuros profesionales. Vimos en este proyecto la oportunidad de devolverle a la sociedad, aunque sea algo muy pequeño, la posibilidad que nos brindó de poder realizar nuestros estudios en una universidad pública, y sin lugar a dudas no solo formarnos como profesionales, sino también el hecho de contribuir a nuestra formación como personas que somos, y que como tales formamos parte de la sociedad en donde está inmersa nuestra universidad.

11 Bibliografía.

- <https://proyectosagiles.org/>
- <http://www.sintecho.org.ar/>
- Material de estudio de la cátedra “Métodos Ágiles” – UTN - FRSF.
- <https://developers.google.com/maps/?hl=es>
- Curso - Webinar: “Diseño Responsivo: Respondiendo a dispositivos, resoluciones y experiencias.”
- <http://www.wampserver.com/en/>
- Material de estudio de la cátedra “Ingeniería de Software” – UTN - FRSF.

Referencias a figuras utilizadas dentro del informe:

- Figura 2: <http://www.lsi.us.es/cursos/cursoweb/cap0801.html>
- Figura 3: <http://blog.ayzweb.com/tutorial/dreamweaver-vs-mvc-modelo-vista-controlador-o-framework>
- Figura 4: <http://blog.tednologia.com/mvc-en-php/>
- Figura 5:
<https://riunet.upv.es/bitstream/handle/10251/34594/memoria.pdf?sequence=1>
- Figura 6: <http://www.agilemarketing.net/agile-marketing-sprint-planning/>

12 Anexo 1: Diseño Web Responsivo o Responsive WebDesign.

Responsive Web Design hace referencia a una serie de técnicas que permiten a las páginas web adaptarse al medio a través del cual un usuario está accediendo a la misma. Los tamaños de pantalla cambian según el medio con el que se accede (no es lo mismo una pantalla de un iPhone que la de un monitor panorámico de sobremesa) pero el usuario cada vez más exige que su experiencia usando la web sea la óptima en cada caso concreto.

Diseño Web Responsivo es un concepto que combina CSS, CSS3 y JavaScript para crear diseños web fluidos y adaptables que se pueden ampliar, contraer, reorganizar o eliminar el contenido en función del tamaño de la pantalla que utilice el usuario.

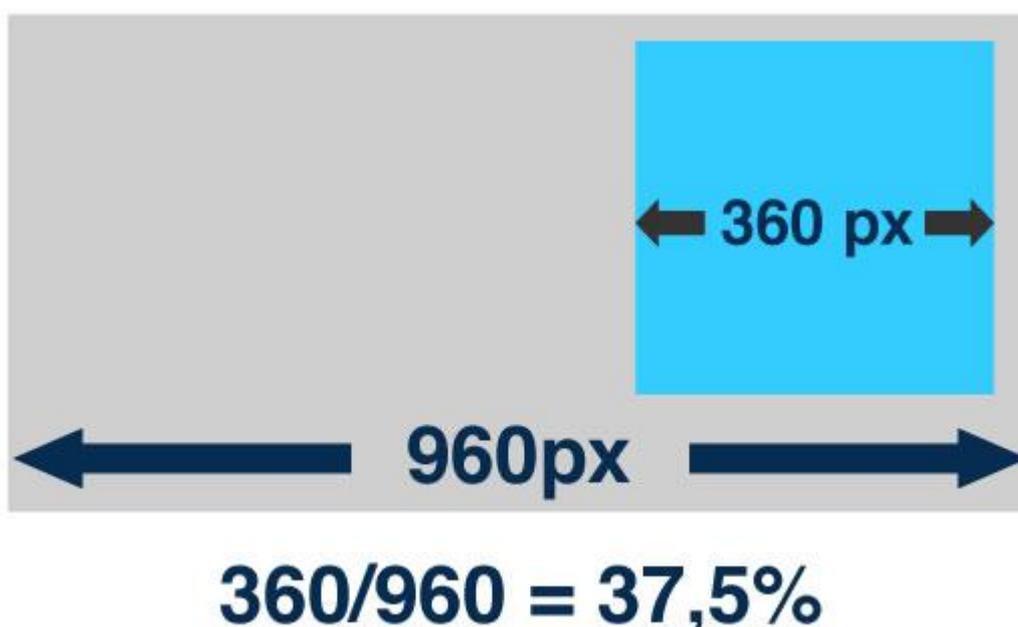
Partiendo de la base de que el diseño de una web, para que funcione, tiene que estar centrada en el usuario (y no en el diseñador, en el programador o en el dueño de la web), es importante que la experiencia que tiene el usuario con nuestra web sea lo más placentera posible con independencia de qué medio esté usando para verla. Es por ello que últimamente tantos sitios webs están utilizando lo que se llama Responsive Web Design o Diseño Web adaptable.

En vez de desarrollar sitios web diferentes para los dispositivos con distintos tamaños de pantalla y capacidades, una web con un diseño Responsivo (adaptable) reacciona de manera flexible para ser visualizada de forma óptima en cualquier pantalla.



Primer paso: El diseño fluido

El principal concepto en el que se apoya el Diseño Web Adaptable es en abandonar los anchos fijos de nuestra web. Estos deberán ser fluidos. En lugar de diseñar nuestra web basándonos en valores fijos (por ejemplo width: 960px), el diseño fluido está pensado en términos de proporciones. De esta manera cuando veamos nuestra web a través de la pequeña pantalla de un móvil todos los elementos de la web se harán más pequeños guardando la proporción entre ellos. Por ejemplo, para saber ahora el ancho de un elemento tendremos que dividir el ancho inicial del mismo entre el ancho del elemento "padre", por llamarlo de alguna manera sencilla. Pongamos que tenemos por ejemplo esta estructura:



En este ejemplo partíamos de unos valores fijos: un contenedor de 960 pixels y dentro del mismo un elemento de 360 pixels de ancho. Si dividimos el segundo entre el primero y multiplicamos el resultado por 100 obtendremos el valor de 37,5%, que será el ancho que aplicaremos a dicho elemento. Es decir, el ancho del elemento interior será siempre el 37,5% del ancho del primero. De esta forma cuando el ancho del elemento "padre" se adapta, **todos los anchos de los elementos interiores varían en función de su porcentaje**. Ahora el elemento interno, en la hoja de estilos, tendrá en lugar de un width="360px" un width="37,5%".

Lo mismo haremos con los **tamaños de las fuentes** (por ejemplo, si el tamaño general es del 100%, que equivale a 16px, y tenemos un título de 22px, su nuevo tamaño será de $22/16 = 1.375em$).

¿Pero, qué pasa con las imágenes u otros elementos que tienen un ancho fijo? Podemos adaptar su ancho así:

img, video, object { max-width:100%; }

De esta manera su ancho **nunca excederá** del ancho del elemento que la contiene. Y si dicho elemento cambia de ancho, también lo hará la imagen en todos los navegadores modernos. IE7 e IE6 no lo soportan. Para estos navegadores lo mejor es incluir en su hoja de estilos específica:

img, video, object {width:100%; }

Esta regla es completamente distinta de la anterior: Ahora decimos que la imagen (por ejemplo) siempre tendrá el mismo ancho de su contenedor. Es por ello por lo que hay que tener cuidado sobre qué elemento se aplica.

Esto está muy bien hasta que nos encontramos con anchos de pantalla realmente pequeños (por ejemplo un móvil). Si tenemos una web con tres columnas, muchos botones, menú horizontal a la derecha del logo, etc. Al comprimir tanto el tamaño de la pantalla, por mucho que los anchos sean fluidos, puede acabar todo en un caos. Es probable que tengamos que prescindir de ciertos elementos de la web o situarlos en un lugar diferente. Para ello utilizaremos los **Media Queries**.



Segundo paso: Los Media Queries

Como decíamos ningún diseño escala de manera adecuada cuando cambia el contexto para el que fue pensado. Los Media Queries forman parte de CSS3 e inspeccionan las características físicas del medio que va a mostrar nuestro diseño (no olvidemos que “query” equivale a “pregunta”, es como preguntar: ¿qué medio se está usando?). Si las características del medio utilizado por el usuario están dentro de un condicional establecido con los Media Queries, se aplicarán una serie de instrucciones CSS contenidas dentro del mismo, de esta manera cuando nuestro diseño fluido cambia de tamaño podremos aplicar una serie de instrucciones CSS pensadas en exclusiva para ese nuevo tamaño. Vamos a ver un ejemplo.

El ancho de pantalla actual del iPhone es de 320px. Vamos a suponer que para ese ancho nuestro diseño fluido presenta una serie de dificultades (puede ser desde cambiar el logo, eliminar una columna, cambiar la organización de los elementos de la pantalla, etc...). Dentro de nuestra hoja de estilos haríamos:

```
@media screen and (max-width: 320px) {  
  /* Aquí van las reglas CSS necesarias */  
}
```

Como se puede ver la instrucción se compone de dos partes: el tipo de medio utilizado (o Media Type, en este caso “screen”, que agrupa a todos los medios que se ven vía una pantalla) combinándolo mediante un “and” con el Media Query (*max-width: 320px*). Entonces nos realizamos la siguiente pregunta: ¿Es un medio con pantalla y tiene un ancho de 320px o menor? Entonces le aplicamos los estilos situados entre los corchetes correspondientes.

Podemos empezar desde este ancho e ir subiendo a otras posibles opciones. Algunos autores recomiendan optimizar estos anchos de pantalla:

320px
480px
600px
768px
900px
1200px

¿Qué navegadores soportan los Media Query?

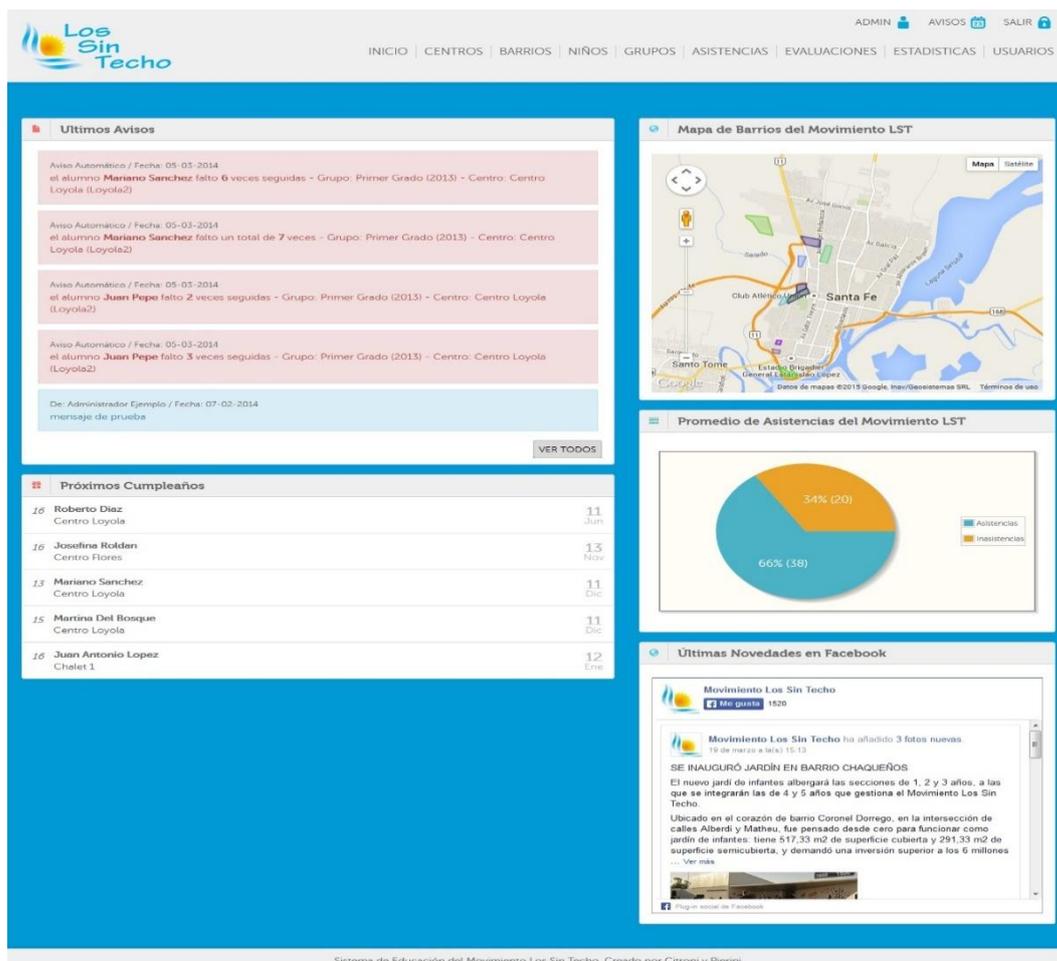
En general todos los navegadores modernos lo soportan. Afortunadamente hay soluciones para aquellos navegadores que no lo soportan utilizando Javascript, por ejemplo respond.js. Esto es, *“respond.js proporciona un script rápido y ligero (3kb minified / 1kb gzipped) que permite utilizar diseños web adaptables en navegadores que no soportan CSS3 Media Queries – en concreto Internet Explorer 8 e inferiores”*.

13 **Anexo 2: Diseño final de la interfaz implementada en el sistema de educación.**

- MenúLogin



- Menú Principal



- Listado de centros educativos.

Los Sin Techo

ADMIN AVISOS 23 SALIR

INICIO | CENTROS | BARRIOS | NIÑOS | GRUPOS | ASISTENCIAS | EVALUACIONES | ESTADISTICAS | USUARIOS

Id	Nombre	Barrio	Acciones
1	Centro Loyola	La Loma	
2	Centro Flores	Centenerio	
3	Centro Edu Hornos	Los Hornos	
4	Chalet 1	Chalet	

Buscar:

Primero Anterior 1 Siguientes Último

NUEVO CENTRO

- Alta de centros educativos.

NUEVO CENTRO DE EDUCACIÓN

NOMBRE DEBE INGRESAR UN NOMBRE

BARRIO AL QUE PERTENECE SELECCIONE UN BARRIO DEBE SELECCIONAR UN BARRIO

GUARDAR CERRAR

- Listado de Barrios.

Id	Nombre	Acciones
1	La Loma	[Search] [Edit] [Delete]
2	Centenerio	[Search] [Edit] [Delete]
3	Los Hornos	[Search] [Edit] [Delete]
4	Chalet	[Search] [Edit] [Delete]
5	Barranquitas Sur	[Search] [Edit] [Delete]
6	Villa Hipódromo	[Search] [Edit] [Delete]
7	Villa del Parque	[Search] [Edit] [Delete]
8	Villa Oculta	[Search] [Edit] [Delete]
9	San Lorenzo	[Search] [Edit] [Delete]
10	San Pantaleón	[Search] [Edit] [Delete]

- Vista de Barrios General.

- Servicios por barrio.

✕
MODIFICAR BARRIO

☰ GENERAL
SERVICIOS
COLEGIOS
IGLESIAS
BARRIOS LIMITROFES

NOMBRE	ESTADO	OBSERVACION	ACCIONES
LUZ	MALO	Muchos Colgados, Pocos Medidores Comun	
RECOLECCION DE BASURA	INEXISTENTE	No hay recoleccion municipal, los vecinos se €	
HIGIENE	MALO	la situacion es critica, muchos perros y ratas, y	
CLOACAS	INEXISTENTE		
TELEVISION	MALO	Solo tv por cable, varios vecinos usan el mism	
INTERNET	MALO	a traves de modems o celulares, no hay ciber,	

AGREGAR SERVICIO

GUARDAR
CERRAR

- Barrios limítrofes.

✕
MODIFICAR BARRIO

☰ GENERAL
SERVICIOS
COLEGIOS
IGLESIAS
BARRIOS LIMITROFES

BARRIOS: SELECCIONE UN BARRIO

AGREGAR BARRIO COMO LIMITROFE

ID	BARRIO	ACCIONES
1	LA LOMA	

GUARDAR
CERRAR

- Listado de alumnos.

Id	Nombre	Apellido	DNI	Edad	Cumpleaños	Centro	Colegio	Acciones
1	Juan	Perez	12345678	9	31/01	Centro Loyola	Padre Catena	
2	Roberto	Diaz	22222222	11	11/06	Centro Loyola	Reimundo Peña	
3	Mariano	Sanchez	33333333	9	11/12	Centro Loyola	Patricio Cullen	
4	Martina	Del Bosque	44444444	8	11/12	Centro Loyola	Monseñor Zaspé	
5	Josefina	Roldan	35673276	7	13/11	Centro Flores	Monseñor Zaspé	
6	Juan Antonio	Lopez	41235322	10	12/01	Chalet 1	Patricio Cullen	
7	Santiago	Cueva	40234239	6	11/03	Centro Edu Hornos	Reimundo Peña	
8	Jeremias	Sosa	42332421	8	08/09	Centro Loyola	Monseñor Zaspé	
9	Alejandro	Veliz	41234429	6	08/10	Centro Flores	Dr Rodolfo Freyre	
10	Juan Jose	Romero	43449928	7	04/08	Centro Edu Hornos	Falucho	

Buscar: Primero Anterior 1 2 Siguiete Último

- Vista de información sobre alumnos.

MODIFICAR ALUMNO

INFORMACION GENERAL | GRUPOS

NOMBRE: CENTRO AL QUE ASISTE:

APELLIDO: COLEGIO AL QUE ASISTE:

D.N.I.: FECHA DE NACIMIENTO:

FAMILIA SARAMI

- Listado de grupos.

Id	Nombre	Centro	Año	Responsable	Acciones
1	Primer Grado	Centro Loyola	2013	Margarita Dominguez	
2	Segundo Grado	Centro Loyola	2013	Margarita Dominguez	
3	Tercer Grado	Centro Edu Hornos	2013	Margarita Dominguez	
4	Segundo Grado	Centro Loyola	2014	Margarita Dominguez	
5	Tercer Grado	Centro Loyola	2015	Adriana Castro	
6	Primer Grado	Centro Flores	2014	Adriana Castro	

- Información sobre cada uno de los grupos.

- Alumnos pertenecientes a un grupo en particular.

MODIFICAR GRUPO
✕

DATOS DEL GRUPO
ALUMNOS

ALUMNO:

ID-ALUMNO	ALUMNO	ACCIONES
1	JUAN PEREZ	
4	MARTINA DEL BOSQUE	
5	JOSEFINA ROLDAN	
2	ROBERTO DIAZ	
3	MARIANO SANCHEZ	
9	ALEJANDRO VELIZ	

- Interfaz de carga de asistencias.

🔍 Registrar asistencia de Alumnos

FILTRO

GRUPO

PRIMER GRADO (2013) - CENTRO LOYOLA
 SEGUNDO GRADO (2013) - CENTRO LOYOLA
 SEGUNDO GRADO (2014) - CENTRO LOYOLA
 TERCER GRADO (2015) - CENTRO LOYOLA
 PRIMER GRADO (2014) - CENTRO FLORES
 TERCER GRADO (2013) - CENTRO EDU HORNO

FECHA

LISTADO DE ASISTENCIA

PRESENTE	ID	NOMBRE	APELLIDO	DNI	EDAD	CUMPLEAÑOS	CENTRO
<input type="checkbox"/>	1	JUAN	PEREZ	12345678	9	31/01	CENTRO LOYOLA
<input type="checkbox"/>	4	MARTINA	DEL BOSQUE	44444444	8	11/12	CENTRO LOYOLA
<input type="checkbox"/>	5	JOSEFINA	ROLDAN	35673276	7	13/11	CENTRO FLORES
<input type="checkbox"/>	2	ROBERTO	DIAZ	22222222	11	11/06	CENTRO LOYOLA
<input type="checkbox"/>	3	MARIANO	SANCHEZ	33333333	9	11/12	CENTRO LOYOLA
<input type="checkbox"/>	9	ALEJANDRO	VELIZ	41234429	6	08/10	CENTRO FLORES

SELECCIONAR TODOS

Citroni - Pierini

Página 58

- Interfaz consulta de asistencias.

Consultar Asistencias de los Alumnos

FILTRO

CENTRO: CENTRO LOYOLA DESDE: 01-12-2013

GRUPO: PRIMER GRADO(2013) HASTA: 31-12-2013

CONSULTAR

ASISTENCIAS REGISTRADAS

ID - ALUMNO	11/12/13	12/12/13	13/12/13	14/12/13	15/12/13	16/12/13	18/12/13	19/12/13	20/12/13	21/12/13
1 - JUAN PEREZ	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗
4 - MARTINA DEL BOSQUE	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓
5 - JOSEFINA ROLDAN	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
2 - ROBERTO DIAZ	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
3 - MARIANO SANCHEZ	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗
9 - ALEJANDRO VELIZ	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗

IMPRIMIR ASISTENCIAS

- Interfaz consulta evaluaciones.

Evaluacion de los Alumnos

FILTRO

BARRIO: LA LOMA GRUPO: PRIMER GRADO(2013)

CENTRO: CENTRO LOYOLA TIPO DE EVALUACION: EVALUACION FINAL

BUSCAR ALUMNOS

ALUMNOS A EVALUAR

ID	NOMBRE	APELLIDO	EVALUADO	ACCIONES
1	JUAN	PEREZ	SI	✍
4	MARTINA	DEL BOSQUE	NO	✍
5	JOSEFINA	ROLDAN	NO	✍
2	ROBERTO	DIAZ	SI	✍
3	MARIANO	SANCHEZ	NO	✍
9	ALEJANDRO	VELIZ	NO	✍

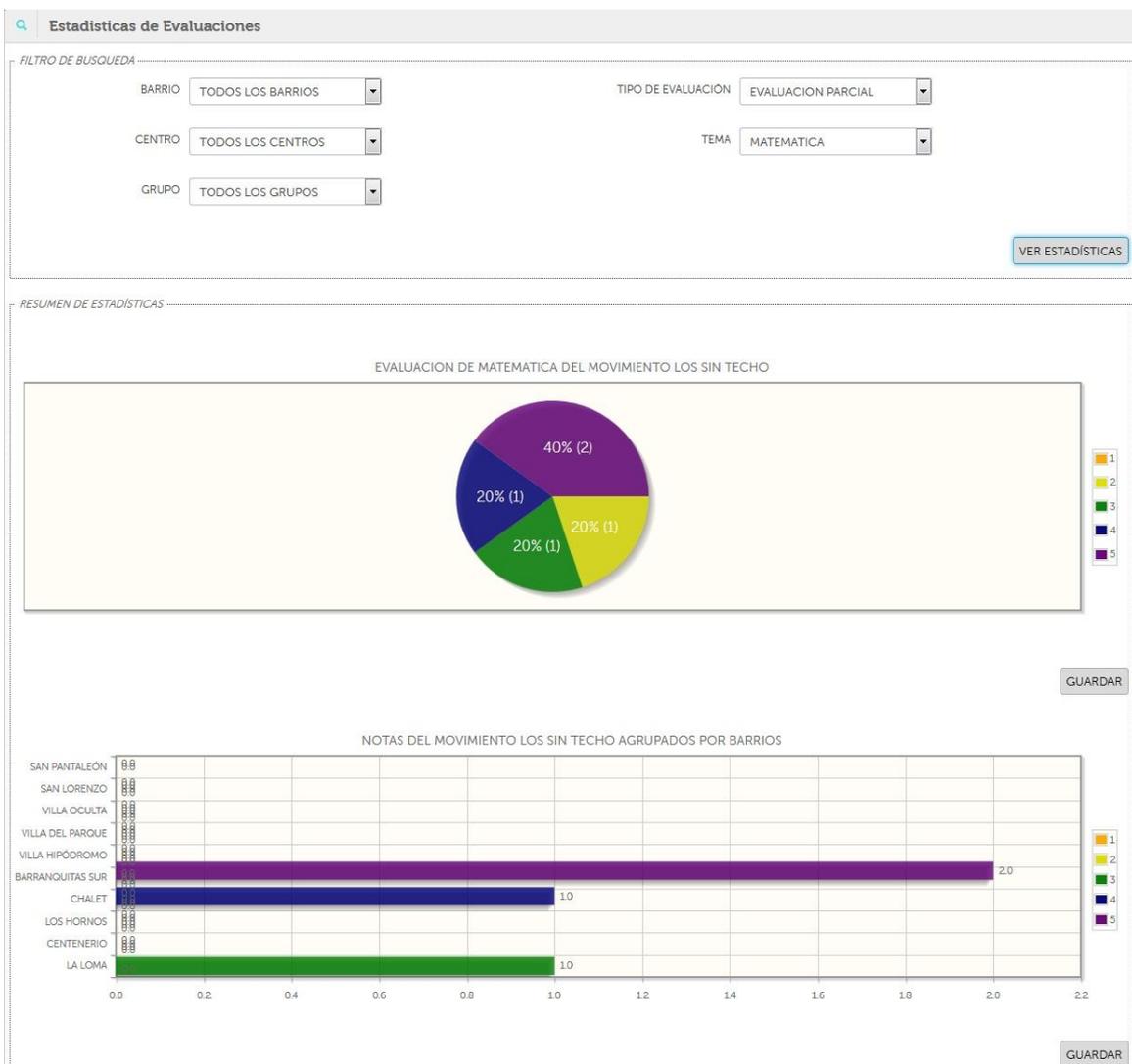
- Interfaz carga de evaluaciones.

EVALUAR ✕

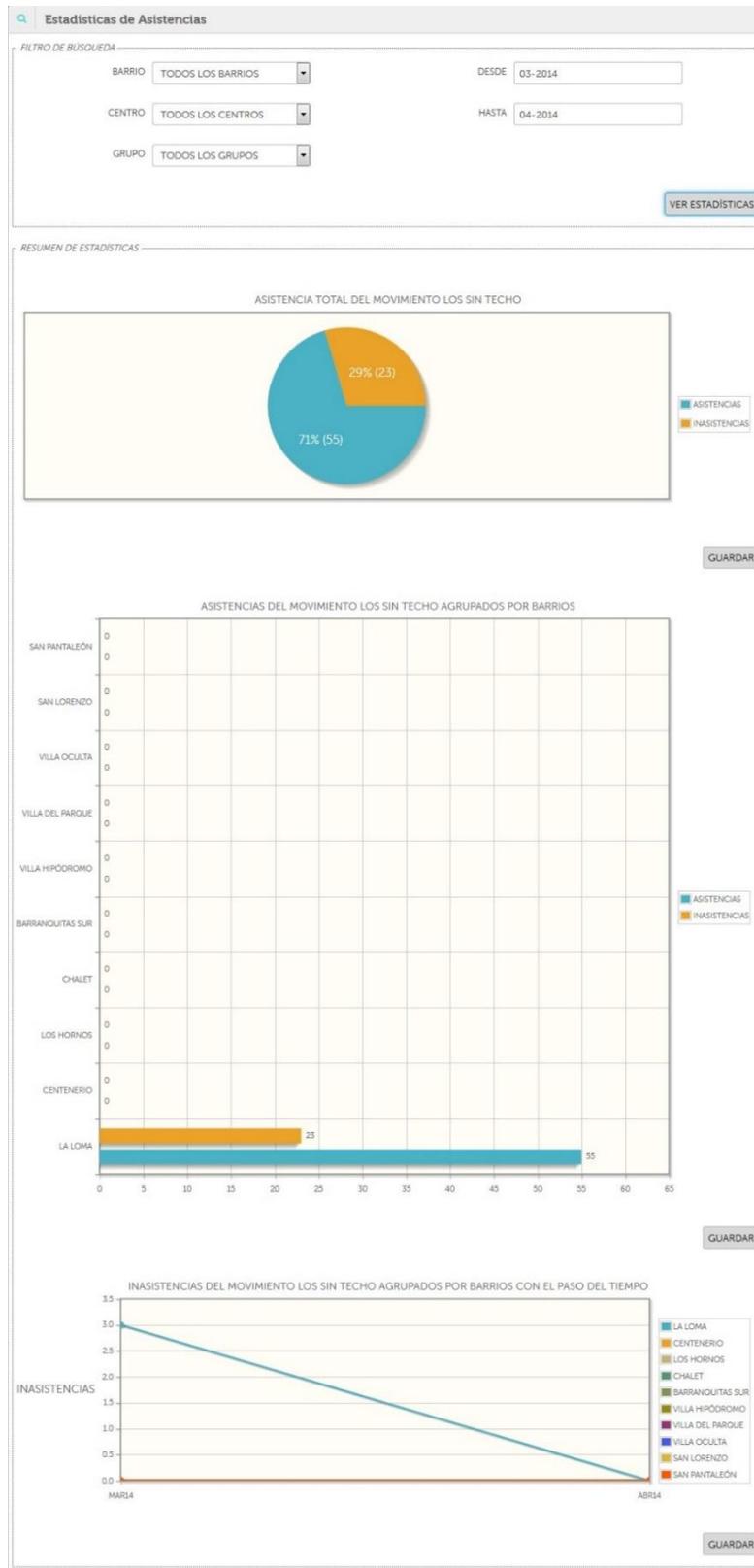
NOTAS DE ROBERTO DIAZ

SUMA: <input type="text" value="5"/>	RESTA: <input type="text" value="4"/>	MULTIPLICACION: <input type="text" value="5"/>
DIVISION: <input type="text" value="5"/>	PROBLEMAS: <input type="text" value="2"/>	SUSTANTIVOS: <input type="text" value="2"/>
ADJETIVOS: <input type="text" value="4"/>	VERBOS: <input type="text" value="1"/>	ORACIONES: <input type="text" value="1"/>
LECTURA: <input type="text" value="MUY BUENO"/>	ESCRITURA: <input type="text" value="MUY BUENO"/>	HORA: <input type="text" value="BUENO"/>
CONDUCTA: <input type="text" value="BUENO"/>		

- Interfaz sobre estadísticas de evaluaciones.



- Interfaz sobre estadísticas sobre asistencias.



- Listado de usuarios cargados en el sistema.

Id	Usuario	Nombre	Apellido	Tipo	Acciones
1	mardo	Margarita	Dominguez	Maestro	[Iconos de búsqueda, edición, eliminación]
2	admin	Ruben	Benitez	Administrador	[Iconos de búsqueda, edición, eliminación]
3	acastro	Adriana	Castro	Maestro	[Iconos de búsqueda, edición, eliminación]
4	aguparedes	Agustin	Paredes	Directivo	[Iconos de búsqueda, edición, eliminación]

- Listado de avisos creados en el sistema.

Id	Fecha	Mensaje	Remitente	Visto	Acciones
1	2014-02-07	mensaje de prueba	Ruben Benitez	No	[Iconos de edición, eliminación]

Id	Fecha	Mensaje	Alumno	Visto	Acciones
2	2014-03-05	el alumno Mariano Sanchez falto 6 veces seguidas - Grupo: Primer Grado (2013) - Centro: Centro Loyola (Loyola2)	Mariano Sanchez	No	[Icono de eliminación]
3	2014-03-05	el alumno Mariano Sanchez falto un total de 7 veces - Grupo: Primer Grado (2013) - Centro: Centro Loyola (Loyola2)	Mariano Sanchez	No	[Icono de eliminación]
4	2014-03-05	el alumno Juan Pepe falto 2 veces seguidas - Grupo: Primer Grado (2013) - Centro: Centro Loyola (Loyola2)	Santiago Cueva	No	[Icono de eliminación]
5	2014-03-05	el alumno Juan Pepe falto 3 veces seguidas - Grupo: Primer Grado (2013) - Centro: Centro Loyola (Loyola2)	Santiago Cueva	No	[Icono de eliminación]

- Interfaz de edición de avisos.

14 Anexo 3. Visualización y Delimitación Geográfica de los Barrios a Través de Google Maps

Google Maps es un servicio de Google que ofrece imágenes vía satélite de todo el planeta, combinadas, en el caso de algunos países, con mapas de sus ciudades, lo que unido a sus posibilidades de programación abierta ha dado lugar a diversas utilidades ofrecidas desde numerosas páginas web.

Google Maps dispone de un amplio conjunto de APIs (Interfaces de Programación de Aplicaciones) que te permiten trasladar la gran funcionalidad y la utilidad diaria de Google Maps a tu propio sitio web y a tus propias aplicaciones, así como superponer tus datos.

En nuestro caso utilizamos la Versión 3 del API de JavaScript de Google Maps, la cual con un simple código Javascript permite visualizar mapas y personalizarles cierta información de Interés.

A continuación vamos a describir los pasos para visualizar un simple barrio en nuestra aplicación:

Inicialmente disponemos de los vértices geodésicos del barrio, las cuales son simples coordenadas geográficas (latitud y longitud) que representan la ubicación exacta de cada punto que delimita el barrio (entendiéndose este como un polígono dentro de un mapa).

Esta información ya disponible, la tenemos almacenada en La Base de Datos:

bar_id	1
bar_nombre	La Loma
bar_mapa	-60.73595688091992,-31.59310406950754 -60.733455920922815,-31.60214940137516 -60.739393374662804,-31.600771063899806 -60.74244162100255,-31.59560229836722 -60.74525130893308,-31.59345527268445 -60.74548986734227,-31.592235974148558 -60.73595688091992,-31.59310406950754
bar_tipo_calle	(Null)
bar_manzanas	(Null)
bar_observacion	(Null)

En nuestro código HTML primero debemos incluir el API:

```
<script src="//maps.googleapis.com/maps/api/js?sensor=false"
  type="text/javascript"></script>
```

Luego lo único que debemos crear es el DIV correspondiente, en donde el javascripts se encargara de insertar el mapa generado:

```
<div id="mlst_mapa"></div>
```

Con los Puntos ya disponibles lo único que debemos hacer es generar el siguiente código Javascript:

```
setTimeout(function() {

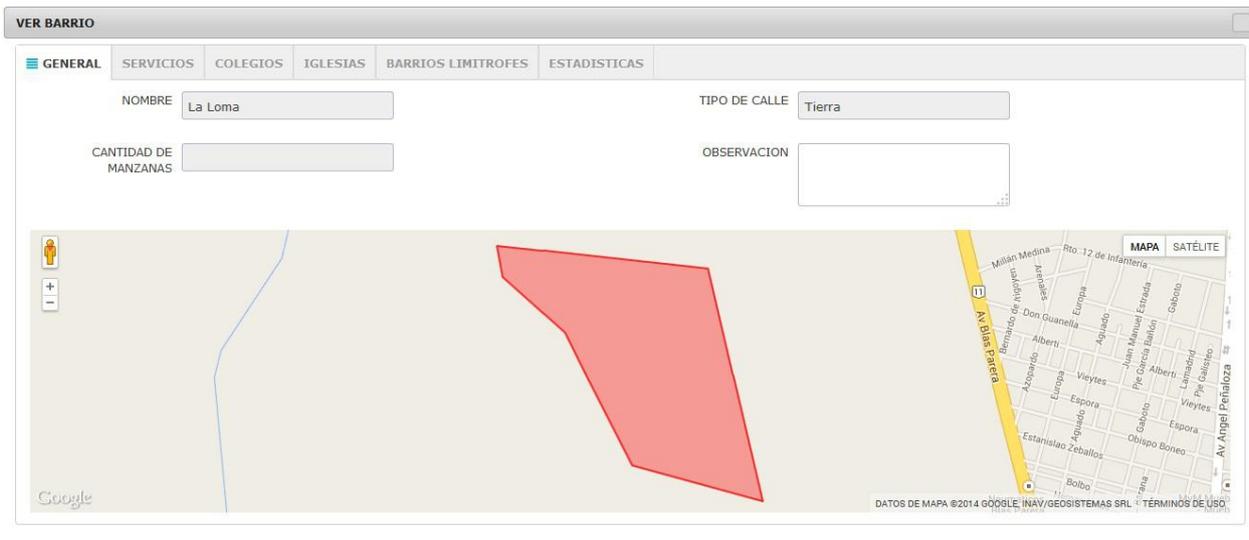
  // Mapa de Santa Fe
  var map = new google.maps.Map(document.getElementById('mlst_mapa'), {
    zoom: 12,
    center: new google.maps.LatLng(-31.6207752, -60.7012926),
    mapTypeId: google.maps.MapTypeId.ROADMAP
  });

  // Barrio La Loma
  var barriol = new google.maps.Polygon({
    paths: [
      new google.maps.LatLng(-31.59310406950754, -60.73595688091992),
      new google.maps.LatLng(-31.60214940137516, -60.733455920922815),
      new google.maps.LatLng(-31.600771063899806, -60.739393374662804),
      new google.maps.LatLng(-31.59560229836722, -60.74244162100255),
      new google.maps.LatLng(-31.59345527268445, -60.74525130893308),
      new google.maps.LatLng(-31.592235974148558, -60.74548986734227),
      new google.maps.LatLng(-31.59310406950754, -60.73595688091992),
    ],
    strokeColor: '#1C000E',
    strokeOpacity: 0.0,
    strokeWeight: 2,
    fillColor: '#1C000E',
    fillOpacity: 0.35
  });

  barriol.setMap(map);
}, 200);
```

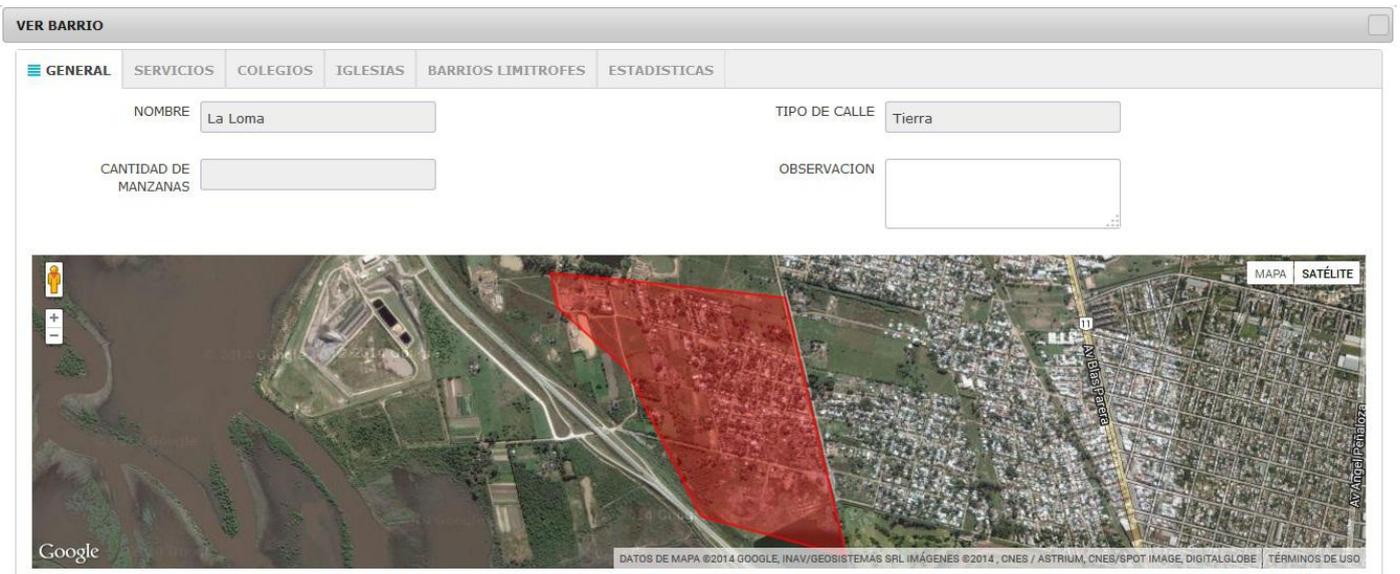
Como se observa en el código, por un lado se crea un nuevo Mapa, y por otro lado se crea un nuevo Polígono (que representa el barrio). Finalmente solo queda asociar el polígono al Mapa.

Como resultado final, la visualización del mapa es la siguiente:



Primer ejemplo de mapa generado con los puntos delimitantes de un barrio.

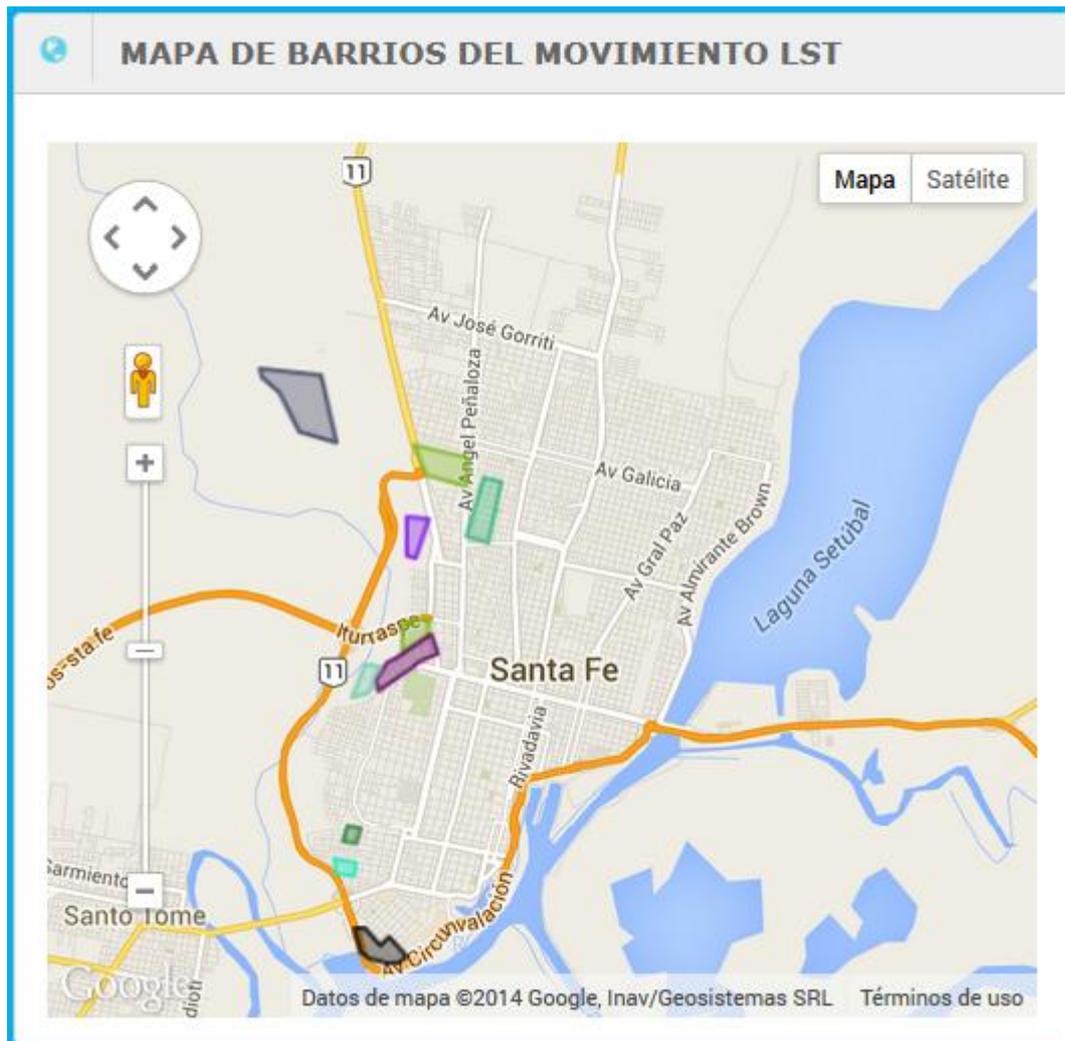
En este caso podemos ver que se creó el polígono sobre el mapa de Google Maps, pero casualmente en este caso, Google Maps no tiene las calles creadas sobre este barrio. Para entender mejor la ubicación del barrio, podemos cambiar la Vista que Google nos brinda, por una vista satelital:



Segundo ejemplo de mapa generado con los puntos delimitantes de un barrio.

Aquí se puede entender realmente como está conformado el Barrio y cuales son por ejemplo las calles que lo delimitan.

Conociendo esta API de Google, se puede mejorar la utilidad, visualizando por ejemplo, varios Barrios en un mismo Mapa. Simplemente con un código similar a lo ya visto, nuestro mapa puede verse de la siguiente forma:



Delimitación de barrios dentro de la ciudad de Santa Fe.

15 ANEXO 4. Instalación del software en entorno Windows Server

Requerimientos técnicos del servidor:

- Acceso a Internet.

Requerimiento de Software base:

- Apache >= 2.2

- php >= 5.3

- mySql >= 5.5

(Todos estos componentes se obtienen en un solo paquete de WampServer, específicamente la versión 2.2 del mismo)

Link de Descarga WampServer 2.2:

<https://sourceforge.net/projects/wampserver/files/WampServer%202/WampServer%202.2/>

Se debe descargar e instalar este producto, el cual automáticamente instala todos sus componentes.

No es necesario habilitar ningún módulo ni extensión aparte de las que se instalan por defecto en Apache y PHP.

Configuración de Base de Datos:

- Con MySQL ya instalado, se debe crear una Base de Datos y luego importar los Script de BD de la aplicación.

Para ello puedo utilizar el componente que incluye WampServer: phpMyAdmin. Alguna herramienta de Gestión como Workbench o Navicat. o directamente por línea de comando.

- El nombre de la nueva Base de Datos es indistinto y el charset de la misma debe ser UTF-8.
- Los Script se deben ejecutar ubicándose en la Base de Datos recién creada, y en el siguiente orden:
aplicacion/db/centroedu.sql
aplicacion/db/script-140728.sql
aplicacion/db/centroedu-datos.sql

Configuración de la aplicación:

- Ubicarse en el directorio "aplicacion/settings"
- Copiar el archivo "config_example.ini" en la misma ubicación y renombrarlo a "config.ini"
- Ingresar al archivo y configurar la conexión a Base de Datos en las siguientes líneas:

[base]

base="" ; Nombre de la Base de Datos creada

odbc="" ; Dominio donde está ubicada, puede ser "localhost" (ya que la aplicación se encuentra en el mismo servidor que la Base de Datos), la IP del servidor, o el nombre del dominio

usuario_odbc = "" ; Usuario de Base de Datos

password_odbc = "" ; Contraseña de Base de Datos

*Nota: La aplicación también puede desplegarse en un entorno Linux, solo basta con descargarse la versión de Apache correspondiente al sistema operativo.
Ambos entornos (tanto Windows como Linux) son compatibles con la aplicación.*

17 ANEXO 6. Diagrama de despliegue.

