

# Universidad Tecnológica Nacional

## Facultad Regional Córdoba

### Ingeniería en Sistemas de Información

### Proyecto Final

## Documentación del Proyecto

#### 5° AÑO 5K2

**Docentes:** Zohil, Julio Cesar Nelson (Adjunto)  
Liberatori, Marcelo Sadi (JTP)  
Jaime, María Natalia (JTP)

**Alumnos:** Bär Coch, Edgardo Ezequiel  
Barrera, Marcos  
Frigerio, Ignacio  
Romero, Leonel  
Toneatto, Lucas Fernando

## Contenido

Plan de Proyecto .....	6
Control de la documentación.....	6
Control de la Configuración.....	6
Histórico de Versiones.....	6
Introducción .....	7
Determinación de un objetivo .....	7
Work Breakdown Structure (WBS).....	7
Análisis de Factibilidad .....	8
Calendarización .....	8
Gestión de Riesgos .....	8
Gestión de Configuración.....	9
Auditorias .....	9
Conclusión .....	10
Gestión de Riesgos .....	11
Control de la documentación.....	11
Control de la Configuración.....	11
Histórico de Versiones.....	11
Introducción .....	12
Identificación de Riesgos.....	12
RIESGOS.....	12
Análisis de Riesgos .....	13
Análisis cualitativo de riesgos .....	13
Análisis cuantitativo de riesgos.....	14
Planificación de respuesta al Riesgo .....	15
Supervisión y Control de Riesgos .....	17
Conclusión .....	19
Plan de gestión de la configuración .....	20
Control de la documentación.....	20
Control de la Configuración.....	20
Histórico de Versiones.....	20
Introducción .....	21
Propósito .....	21

Alcance .....	21
Acrónimos .....	21
Definiciones.....	21
Organización.....	22
Sistema de Gestión de la Configuración.....	22
Personal, Roles y Responsabilidades.....	22
Herramientas.....	22
Procedimientos .....	23
Estimación de tiempo para identificación de Elementos.....	23
Identificar elementos de configuración .....	23
Establecer un sistema de administración de configuración.....	24
Crear o liberar las líneas base .....	25
Seguir las peticiones de cambio .....	26
Controlar los elementos de configuración.....	27
Realizar auditorías de configuración .....	28
Normas de Desarrollo .....	29
Control de la documentación.....	29
Control de la Configuración.....	29
Histórico de Versiones.....	29
Introducción .....	30
Comentarios.....	30
Recomendaciones generales.....	30
Otras recomendaciones .....	32
Sangrías, espacios y líneas en blanco.....	33
CONVENCIÓN DE NOMBRES .....	35
GUÍA DE ESTILO .....	38
APÉNDICE A: NOMENCLATURA DE ESPACIOS DE NOMBRES .....	45
APÉNDICE B: EXCEPCIONES .....	45
APÉNDICE C: COMPATIBILIDAD CON OTROS LENGUAJES .NET.....	45
Plan de testing.....	47
Control de la documentación.....	47
Control de la Configuración.....	47
Histórico de Versiones.....	47

Introducción .....	48
Plan de Testing .....	48
Ciclos de Prueba Manuales .....	48
Ciclos de Pruebas Automáticos .....	49
Tests Exploratorios .....	49
Tests de Sistema.....	50
Tests de Regresión .....	50
Estudio de Impacto Ambiental.....	51
Control de la documentación.....	51
Control de la Configuración.....	51
Histórico de Versiones.....	51
Proyecto GeoParking.....	52
Impacto Ambiental.....	52
Impacto Social .....	53
Conclusión .....	54
Referencias.....	54
Informe de auditoria .....	55
Control de la documentación.....	55
Control de la Configuración.....	55
Histórico de Versiones.....	55
Introducción .....	56
Forma de Trabajo .....	56
Resultados .....	56
Auditoria de documentación.....	56
Auditoria de informes de reunión.....	58
Auditoria de release .....	58
Auditoria de Software, Matriz de Trazabilidad y Product Backlog.....	59
Conclusión .....	60
Product Backlog.....	61
Control de la documentación.....	61
Control de la Configuración.....	61
Histórico de Versiones.....	61
Introducción .....	62

Product Backlog.....	63
Detalle de historias de usuario.....	68
ANEXO A.....	77

# Plan de Proyecto

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	Plan de Proyecto
<b>Referencia:</b>	GeoP_Proyecto_PlanDeProyecto.docx
<b>Autores:</b>	Lucas Toneatto
<b>Fecha:</b>	18/02/2015

### Histórico de Versiones.

<b>Versión</b>	<b>Fecha</b>	<b>Estado</b>	<b>Responsable</b>	<b>Cambios</b>
1.0_DraftA	11/02/2015	Pendiente de Revisión	Lucas Toneatto [autor]	Creación del documento
1.0	18/02/2015	Aprobado	Ignacio Frigerio	Revisión del documento

## Introducción

La planificación de proyectos forma parte de la gestión de proyectos, la cual se vale de cronogramas tales como diagramas de Gantt para planear y subsecuentemente informar del progreso dentro del entorno del proyecto. Es el proceso para cuantificar el tiempo y recursos que un proyecto costará.

La finalidad del planeamiento de proyecto es crear un plan de proyecto que un gestor de proyectos pueda usar para acompañar el progreso de su equipo.

El planeamiento del proyecto no es algo para hacerse solamente una vez al comienzo del proyecto. Observar el progreso de su equipo y actualizar adecuadamente el plan de proyecto debe ser una tarea constante del gerente del proyecto.

A continuación este documento explicará los puntos clave de nuestro plan de proyecto y como fue atacado en el transcurso del mismo.

## Determinación de un objetivo

Una de las primeras y más importantes tareas a la hora de encarar un proyecto, es definir el porqué del mismo. Su objetivo es la base esencial de cualquier proyecto, es quien le da el sentido de existencia al mismo. Si el objetivo, toda planificación futura que deseemos hacer, no estará enfocada hacia ninguna parte y todos nuestros esfuerzos estarán vagando y perderán la efectividad con la que fueron aplicados.

GeoParking nació con un objetivo claro desde el principio y es por eso que la esencia del proyecto se mantuvo de principio a fin. A pesar de que hubo modificaciones e inclusión de nuevas perspectivas que hacían más completa aun nuestra idea, nunca sentimos dudas de lo que queríamos realizar.

El tener un objetivo completamente definido y cierto, nos permitió desarrollarnos con mayor confianza, ya que sabíamos que el producto final iba a ser que todos esperábamos y que iba cumplir con nuestras expectativas.

## Work Breakdown Structure (WBS)

La WBS es una descomposición jerárquica orientada al entregable, del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de éste y crear los entregables requeridos, con cada nivel descendente de la WBS representando una definición con un detalle incrementado del trabajo del proyecto. La WBS es una herramienta fundamental en la gestión de proyectos.

Al trabajar con metodologías ágiles (Scrum), fue un poco complicado para nosotros armar a WBS, la visión diferente que manejan las metodologías ágiles impiden pensar o visualizar componentes a futuro, sino enfocarse en pequeñas partes del sistema, y que los mismos sean funcionales.

De igual forma, logramos estructurar la WBS basándonos en la idea de proyecto que teníamos y los objetivos que el mismo debería cumplir, incluyendo el trabajo y los módulos generales necesarios para el mismo.

## **Análisis de Factibilidad**

El análisis de factibilidad forma parte del proceso de evaluación al cual debe someterse todo nuevo proyecto. En la actualidad en ocasiones se aborda este tema desde un enfoque económico-financiero fundamentalmente, acompañado de un enfoque técnico.

En nuestro caso al análisis lo realizamos enfocándonos en las etapas que tiene el desarrollo de un software, poniendo énfasis en determinar que todo lo ideado, fuera posible de realizar tanto al principio del proyecto como hasta su culminación.

Ya finalizado el proyecto, y revisando lo expuesto en este análisis, llegamos a la conclusión de que los pasos ejecutados al diseñar dicho análisis fueron imperiosos, ya que permitieron asegurarnos de estar realizando algo viable y que podía terminarse de la manera en la que lo habíamos pensado.

## **Calendarización**

La calendarización es seleccionar un modelo de proceso apropiado, identificando tareas de ingeniería de software que es preciso realizar; estimar la cantidad de trabajo y el número de personas; todo esto conociendo la fecha límite y considerando los riesgos.

Se crea una red de tareas de ingeniería del software que permiten tener el trabajo listo a tiempo, asignar responsabilidades a cada tarea, asegurarse de que se realice y adaptar la red conforme los riesgos se vuelvan realidad.

Esta tarea fue la más difícil o complicada para nuestro equipo. La falta de experiencia en proyectos de esta envergadura y el no tener en claro todas las condiciones que pueden afectar el desarrollo de un proyecto, nos llevaron a tener que rever la calendarización un par de veces; sin embargo la mayor parte de lo planificado fue realizado en el tiempo estimado.

El análisis de situaciones particulares, o eventos no programados nos hicieron realizar una modificación importante de la calendarización. El cambio fue pasar de trabajar en Sprint de 15 días a trabajar los dos últimos Sprints del proyecto de 30 días cada uno.

Este cambio nos permitió adaptar la forma de trabajo del equipo a las condiciones que presentábamos al finalizar la etapa de cursado. Factores como el tiempo disponible para juntarnos, y las distancias entre integrantes, fueron claves al momento de decidir dicha modificación. La misma resultó completamente un éxito, elevando la productividad del equipo ya que le permitía a cada integrante trabajar sin presiones de horarios ni traslados, sino en horarios flexibles a cada uno de ellos.

## **Gestión de Riesgos**

La gestión de riesgos es un aspecto fundamental e imprescindible en cualquier proyecto. El no realizarlo puede traer como consecuencia la finalización de un proyecto en forma temprana por no poder enfrentar problemas de manera adecuada, ya que los mismos no fueron previstos y no se cuenta con un plan de acción.

El análisis que nosotros realizamos tuvo los riesgos comunes a otros proyectos de software, tales como entregas fuera de tiempo por falta de capacitación en nuevas tecnologías, malas estimaciones, mala gestión de los mismos.



Pero lo que se destacó en el mismo, fue un riesgo que peligraba la realización del mismo y que no había acción para enfrentarlo en caso de que ese riesgo se hiciera realidad. El mismo apuntaba a que si el proyecto no se llegaba a finalizar para la fecha límite determinada, integrantes del equipo se iba a ausentar por un tiempo prolongado, dificultando que los demás pudieran rendir la materia y recibirse.

Tener presente ese riesgo fue trabajar bajo presión, principalmente en momentos en surgían desviaciones y retrasos, situaciones que no podíamos darnos el lujo de que sucedieran.

## **Gestión de Configuración**

La gestión de configuración es el conjunto de procesos destinados a asegurar la calidad de todo producto obtenido durante cualquiera de las etapas del desarrollo de un Sistema de Información, a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el desarrollo.

La manera en que fue llevada esta gestión nos permitió contar con toda una documentación ordenada y estructura para un mejor aprovechamiento de la misma. El poder manejar el estado de cada uno de ítems de configuración involucrados en el proyecto, nos brindó la posibilidad de información actualizada, y en el momento que fuera necesario, corregir errores por cambios mal implementados.

El control a nivel de código es indispensable en cualquier proyecto que involucre el desarrollo de software. Contar con un sistema de versionado, evita el re trabajo de escribir o pensar como era el código desarrollado anteriormente a una modificación y mantener versiones del sistema estables sin peligro de corromper la estructura con futuros cambios.

## **Auditorias**

Un aspecto importante a destacar en cuando al manejo del proyecto por parte del equipo, fue la decisión de establecer una auditoria durante el transcurso del mismo.

Esta metodología de control que elegimos implementar fue con el objetivo de establecer un punto de control en el proyecto, que nos pudiera dar una visión analítica de cómo nos estábamos desempeñando como equipo de trabajo y como era el estado del proyecto en ese momento.

La experiencia de realizar la auditoria nos planteó un quiebre en la manera de administrar y llevar a cabo el proyecto. A través de las posteriores Retrospectivas, en donde se analizó la información obtenida de la auditoria, se logó ir mejorando el accionar del equipo.

Todo esto nos dejó un gran aprendizaje teniendo en cuenta los errores de planificación que teníamos al igual que en el desempeño particular de cada uno. Esta retroalimentación nos permitió corregir el curso y apuntar nuevamente hacia el objetivo y contar con más herramientas a la hora de tomar decisiones que afectaran el desarrollo del mismo.

## Conclusión

El desempeño del trabajo en equipo, pero principalmente la manera en que se planifica, pueden definir el éxito o fracaso de un proyecto. Los factores que inciden en la realización del mismo son de los más variados, impactando en aspectos tanto técnicos como organizativos.

El tener planes para sobrellevar situaciones que arriesguen la continuidad del proyecto, permiten que el mismo se desarrolle en un cierto ámbito de estabilidad, necesario para el desempeño de quienes trabajan en él.

Tener un control sobre el mismo, sus estados y sus avances, son aspectos muy importantes para la realización exitosa del objetivo para el que fue planteado el proyecto. EL poder retroalimentarse a medida que avanza el proyecto, permite a las personas involucradas contar con más herramientas para encaminar un proyecto hacia el éxito.

# Gestión de Riesgos

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	Gestión de riesgos
<b>Referencia:</b>	GeoP_Proyecto_GestionRiesgos.docx
<b>Autores:</b>	Lucas Toneatto
<b>Fecha:</b>	29/05/2014

### Histórico de Versiones.

Versión	Fecha	Estado	Responsable	Cambios
1.0_DraftA	29/05/2014	Pendiente de Revisión	Lucas Toneatto [autor]	Creación del documento
1.0	31/05/2014	Aprobado	Leonel Romero [Revisor]	

## Introducción

La gestión de riesgos es una parte integral de la dirección del proyecto, siendo un elemento clave en el proceso de la toma de decisiones.

El mejor modo de evitar el fracaso de un proyecto, es la utilización de ciertas herramientas que permitan gestionar los riesgos.

Como parte de la gestión de proyectos, es necesario definir una política de riesgos de la organización, de manera que la identificación y el tratamiento de los riesgos sea consistente y homogéneo en todo el proyecto.

Es necesario gestionar estos riesgos de manera que a su efecto sobre el proyecto sea nulo o mínimo.

La gestión de riesgos constara de cuatro procesos: identificación, análisis, planificación y control de riesgos, que a continuación se pasan a describir en este documento.

## Identificación de Riesgos

Se identifican los riesgos y disparadores asociados del proyecto, que son síntomas o señales de advertencia de que un riesgo ha ocurrido o está a punto de ocurrir. Requiere considerable planificación e investigación utilizando técnicas diversas.

1. Análisis de las hipótesis y escenarios utilizados en la planificación del proyecto.
2. Entrevistas a personas con experiencia en manejo de tecnologías similares.
3. Análisis de debilidades, amenazas, fortalezas, y oportunidades (DAFO). Este análisis ayuda a una mejor comprensión del proyecto y de los riesgos asociados a cada perspectiva del DAFO.

## RIESGOS

- 1) **Mala estimación de los tiempos:** se podrá percatar del mismo a través del avance que registremos y podamos ver en el “BurndownChart”. Si es repetidos sprint se ve una desfase entre lo estimado y el tiempo real necesitado para realizar dichas tareas, estaremos en presencia de una advertencia de la posibilidad de ocurrencia de riesgo.
- 2) **Falta de conocimiento técnico:** una pista para detectar problemas en este ámbito es la demora o complicaciones que nos vaya imponiendo la tecnología utilizada en las distintas etapas del desarrollo, que nos demanden mayor esfuerzo de lo previsto.
- 3) **Problemas o fallas técnicas en el repositorio:** esta situación es muy difícil de predecir ya que esta fuera de nuestro control la ocurrencia del mismo.
- 4) **El sistema no cumple con las expectativas de los usuarios:** esto podrá ser observado al comenzar con las pruebas de implantación del sistema, aunque nuestro sistema no es algo que haya sido solicitado, sino más bien un producto o solución que nosotros le estamos brindando al mismo.

- 5) **Sobreasignación de tareas al equipo:** se podrá ver el progreso del proyecto cuando los integrantes no puedan ir finalizando en tiempo y forma las tareas que se les ha asignado.
- 6) **Falta de compromiso del equipo:** el estado de ánimo, participación y cumplimiento de los trabajos planeados por el equipo serán métricas o pistas de cómo es el compromiso del mismo para la realización del proyecto.
- 7) **Falta de liderazgo en el proyecto:** un disparador para darnos cuenta de esto, es que nadie del grupo sea capaz de encaminar al mismo a través de actividades de liderazgo que nos organicen y hagan el trabajo más eficiente para el cumplimiento de nuestros objetivos.
- 8) **Subestimar entregables de proyecto:** una forma de darnos cuenta es estar en la situación repetida de tener una sobrecarga de trabajo en días cercanos a las entregas del proyecto y tener que en ocasiones que terminarlo pasado las fechas estipuladas.
- 9) **No se pudo implementar el sistema en Android:** se ira evidenciando a medida que a la hora de desarrollar la parte móvil del sistema nos encontremos con contratiempos que no seamos capaces encontrar la solución, o que la misma exceda nuestras capacidades.
- 10) **No se puede Realizar comunicación con Software de Playa de estacionamiento:** esto se verá llegando la hora de implementar el sistema, después de tener desarrollado el módulo de nuestro sistema que se comunicara con el de ellos.

## **Análisis de Riesgos**

El análisis de riesgos puede ser cualitativo o cuantitativo. El análisis de riesgos tiene como objetivo establecer una priorización de los riesgos del proyecto para su tratamiento posterior.

Esta información puede ser utilizada para apoyar decisiones de inicio o cancelación de un proyecto, para realizar asignaciones de recursos, o para la realización de análisis costo-beneficio.

### **Análisis cualitativo de riesgos**

Este proceso evalúa el impacto y la probabilidad de ocurrencia de los riesgos identificados en el proceso anterior usando métodos y herramientas de análisis cualitativo. El riesgo se mide a partir de dos parámetros: probabilidad e impacto. La probabilidad es la posibilidad de que el riesgo pueda ocurrir. El impacto o severidad es el efecto sobre los objetivos del proyecto, caso de materializarse el riesgo.

Todo riesgo viene definido por sus valores de probabilidad e impacto. Si el riesgo puede materializarse en más de una ocasión, aparece un tercer parámetro de medida: la frecuencia, que mide el número de veces que un determinado riesgo puede materializarse a lo largo del proyecto.

Para que este método sea útil y no lleve a conclusiones erróneas es preciso contar con información precisa y no tendenciosa acerca de los riesgos. Los riesgos deben ser adecuadamente entendidos antes de proceder a la determinación de su probabilidad e impacto. Ello implica

examinar: el grado de conocimiento del riesgo, la información disponible, y la calidad e integridad de la información.

Para medir probabilidad e impacto pueden utilizarse escalas numéricas y no numéricas. En nuestro análisis se hará uso de una escala no numérica para medir el impacto de los riesgos sobre los objetivos del proyecto, utilizando los rangos de: **bajo, medio, y alto**.

#	Ámbito	Conocimiento	Contexto	Momento	PROBABILIDAD	IMPACTO	RIESGO
1	PROYECTO	PREDECIBLE	INTERNO	TODOS	ALTA	ALTO	ALTO
2	TECNICO	PREDECIBLE	INTERNO	DESARROLLO	MEDIA	MEDIO	MEDIO
3	TECNICO	IMPREDECIBLE	EXTERNO	DESARROLLO	BAJA	ALTO	ALTO
4	PROYECTO	PREDECIBLE	INTERNO	TODOS	MEDIA	MEDIO	MEDIO
5	NEGOCIO	PREDECIBLE	INTERNO	IMPLEMENTACION	BAJA	BAJA	BAJO
6	PROYECTO	PREDECIBLE	INTERNO	TODOS	BAJA	BAJA	BAJO
7	PROYECTO	PREDECIBLE	INTERNO	TODOS	BAJA	ALTO	MEDIO
8	PROYECTO	PREDECIBLE	INTERNO	PLANEACION	MEDIA	ALTO	ALTO
9	PROYECTO	PREDECIBLE	INTERNO	TODOS	MEDIA	MEDIO	MEDIO
10	TECNICO	PREDECIBLE	INTERNO	IMPLEMENTACION	BAJA	ALTO	ALTO
11	PROYECTO	PREDECIBLE	INTERNO	DESARROLLO	BAJA	BAJA	BAJO
12	TECNICO	IMPREDECIBLE	INTERNO	DESARROLLO	BAJA	MEDIO	BAJO
13	TECNICO	IMPREDECIBLE	EXTERNO	IMPLEMENTACION	ALTA	ALTO	ALTO

VER TABLA COMPLETA DE ANALISIS ACUANTITATIVO EN ANEXO A (\*)

## Análisis cuantitativo de riesgos

Este proceso utiliza técnicas cuantitativas para determinar la probabilidad y el impacto de los riesgos del proyecto. Generalmente se realiza después del análisis cualitativo de riesgos. Entre las herramientas utilizadas para el análisis cuantitativo del riesgo se encuentran:

- Entrevistas: La información recogida de los expertos es tratada estadísticamente a partir de los datos de algún parámetro concreto cuyo riesgo se quiera estimar. Los datos solicitados dependerán del tipo de distribución a emplear. Por ejemplo, si se usa una distribución triangular se solicitarán 3 valores correspondientes a los escenarios pesimista, optimista, y más probable.

- **Análisis de árbol de decisiones:** Se trata de un diagrama que describe una decisión considerando todas las alternativas posibles. Cada rama incorpora probabilidades de riesgos y los costes o beneficios de las decisiones futuras. La resolución del árbol permite determinar cuál es la decisión que produce el mayor valor esperado. El valor esperado o esperanza matemática se define como el sumatorio de probabilidad por costos y beneficios.
- **Otros:** análisis de sensibilidad, simulación (Análisis de Montecarlo).

#	Denominación	Probabilidad	Impacto en días	VME Exposición
1	Mala estimación de tiempos	0,75	360	270
2	Falta de conocimiento técnico	0,6	20	12
3	Problemas o fallas técnicas en el repositorio.	0,2	10	2
4	No se cumple con los plazos de entrega previstos	0,5	20	10
5	El sistema no cumple con las expectativas de los usuarios	0,1	30	3
6	Sobreasignación de tareas al equipo	0,25	60	15
7	Falta de compromiso del equipo	0,15	30	4,5
8	Subestimar el alcance del sistema	0,6	30	18
9	Falta de liderazgo en el proyecto	0,4	15	6
10	No cumplir todas las funcionalidades del producto	0,5	30	15
11	Subestimar entregables de proyecto	0,3	15	4,5
12	No se pudo implementar el sistema en Android	0,3	40	12
13	No se puede Realizar comunicación con Software de Playa de estacionamiento	0,8	30	24
			<b>total</b>	<b>396</b>

## Planificación de respuesta al Riesgo

Una vez analizados y priorizados los riesgos del proyecto, es preciso proceder a su tratamiento, seleccionando para cada riesgo aquella estrategia de respuesta que tenga mayores posibilidades de éxito. Estas estrategias son:

- Eliminación o evitación: Consiste en eliminar la amenaza eliminando la causa que puede provocarla.

- Transferencia: La transferencia del riesgo busca trasladar las consecuencias de un riesgo a una tercera parte junto con la responsabilidad de la respuesta.
- Mitigación: Busca reducir la probabilidad o las consecuencias de sucesos adversos a un límite aceptable antes del momento de activación. Es importante que los costos de mitigación sean inferiores a la probabilidad del riesgo y sus consecuencias.
- Aceptación: Esta estrategia se utiliza cuando se decide no actuar contra el riesgo antes de su activación. La aceptación puede ser activa o pasiva.

Los planes de mitigación que a continuación se detallaran abarcaran el **80%** de los riesgos que hemos calculado ya que hay una situación que involucra otras y que conjuntas significan un importante impacto que no podemos dejar de suceda por las características y el marco en el que el equipo se ha planteado para el desarrollo del proyecto.

### **MALA ESTIMACION DE TIEMPOS**

- Para este riesgo hemos decidido tomar la postura de “mitigar” el mismo, ya que la ocurrencia del mismo en cualquier etapa del proyecto podría impactar de manera muy desfavorable y haría peligrar de forma importante el desarrollo del proyecto.
- **Plan de Acción**: para reducir al máximo la posibilidad de ocurrencia de este riesgo el equipo ha decidido implementar un plan de acción que consiste en:
  - i. Verificar por Sprint el avance y estado del proyecto.
  - ii. Control constante de inconvenientes planteados por el equipo.
  - iii. Como medida principal, tratar en lo posible de no trasladar ese tiempo mal estimado más allá del siguiente sprint, es decir, que si la situación lo requiere, el equipo deberá trabajar más horas de lo pactado para revertir la situación.

### **FALTA DE CONOCIMIENTO TECNICO:**

- Para este riesgo hemos decidido tomar la postura de “mitigar” el mismo, ya que la ocurrencia del mismo impacta de lleno en el aspecto más importante en el que se basa nuestro proyecto y sistema que es la tecnología.
- **Plan de Acción**: las medidas a tomar serán las siguientes:
  - i. Se realizara la capacitación pertinente en cada tecnología que se vaya a utilizar en el desarrollo del sistema.
  - ii. Lograr tener al menos un “experto den las tecnologías a utilizar, el cual podrá solucionar o evacuar las dudas del equipo de la forma más rápida y eficiente.



### **NO SE PUEDE IMPLEMENTAR EL SISTEMA EN ANDROID:**

- Para este riesgo hemos decidido tomar la postura de “mitigar” y “aceptar” el mismo, ya que la ocurrencia del mismo afectara el desarrollo de uno de los módulos principales del sistema, especialmente el “Core” de mismo.
- **Plan de Acción:** las medidas a tomar serán las siguientes:
  - i. Desarrollaremos un plan alternativo que nos permita tener una segunda opción a la hora de desarrollar el modulo Móvil. Para lograrlo deberemos obtener información suficiente de otros lenguajes alternativos a Android que nos permitan realizar las mismas funcionalidades. Se seleccionara el más adecuado para ser respaldo ante la ocurrencia de este riesgo. Todo esto es teniendo en cuenta la postura de “Aceptar” el riesgo.
  - ii. Para la mitigación del riesgo, llevaremos un control especial sobre las etapas de desarrollo de la aplicación Móvil. El mismo constara de observar tiempo, esfuerzo, inconvenientes surgidos y analizar la viabilidad de seguir con el lenguaje de programación u optar por el lenguaje de respaldo.

### **NO SE PUEDE REALIZAR COMUNICACIÓN CON SOFTWARE DE LA PLAYA DE ESTACIONAMIENTO:**

- a. Para este riesgo hemos decidido tomar la postura de “mitigar” y “aceptar” el mismo
- b. **Plan de Acción:** las medidas a tomar serán las siguientes:
  - i. Para mitigar el riesgo, en etapas cercanas a la implementación del sistema se comenzaran a realizar trámites para poder establecer dialogo con una de las empresas fabricantes de sistemas para playas de estacionamiento, a través de acuerdos lograr una alianza estratégica que nos permita la implementación de nuestro servicio en sus sistemas.
    - 1. Cuando hablamos de aceptar el riesgo, nos referimos a implementar el sistema sin la funcionalidad de disponibilidad, en tiempo real de las plazas de estacionamiento. Como medida extraordinaria planteamos dos alternativas:
      - a. Seguir las tratativas con otras empresas fabricantes de sistemas para playas de estacionamiento.
      - b. Comenzar un proyecto propio para el desarrollo de un sistema de gestión de playa de estacionamiento que permita la comunicación con el sistema ya desarrollado.

## **Supervisión y Control de Riesgos**

Este proceso se ocupa del seguimiento de los riesgos identificados de manera que los planes de riesgos son ejecutados por los responsables asignados de la supervisión de los riesgos residuales, de la aparición de disparadores que indican que algún riesgo está a punto de producirse, de la

revisión de la priorización de riesgos realizada, y de la identificación de nuevos riesgos que pudieran presentarse.

El instrumento más potente de control de riesgos son las revisiones de proyecto. En toda reunión y revisión de proyecto debiera haber un punto de la agenda dedicado al tratamiento de los riesgos, donde se revisarán todos los puntos anteriores.

Otras herramientas de control de riesgo son el análisis de valor de trabajo realizado y la medición de rendimiento técnico que proporcionan datos valiosos sobre desviaciones de los objetivos proyecto.

El desarrollo de software con Scrum hace que la gestión de **riesgos** forme parte del ciclo de vida del proyecto. Esto se debe a su carácter iterativo, el cual conlleva reuniones diarias, reuniones de planificación, reuniones de revisión y retrospectiva. Las diferentes reuniones, permiten que el equipo discuta acerca de impedimentos que podrían convertirse en **riesgos** para la realización del proyecto, lo que permite que el Scrum master pueda mitigarlos inmediatamente. En el caso de las reuniones diarias, la gestión de **riesgos** está más bien enfocada a las historias, mientras que las reuniones de planificación se centran en los **riesgos** del release o sprint como un todo. Por otro lado, en la Retrospectiva el equipo discute los problemas presentados en el sprint e identifican aquellos que pueden llegar a convertirse en **riesgos** para el proyecto. Finalmente, la Reunión de revisión asegura que el producto cumpla con las expectativas de los stakeholders, y si esto no ocurre poder solucionarlo de manera temprana.

Sumado a su carácter iterativo, Scrum es un marco de trabajo que provee flexibilidad en término de los requerimientos. El product backlog es un artefacto que, si bien se elabora en un principio del proyecto, evoluciona a medida de que el producto y el entorno en el que será usado también lo hacen. El mismo es dinámico, cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Esto permite una gestión continua del **riesgo**.

Este marco de trabajo parecería ser apropiado en cuanto al manejo de **riesgos**, ya que los mismos se gestionan desde el principio de ciclo de vida del proyecto y a lo largo de todo su desarrollo. Sin embargo, pensamos que la mayoría de riesgos así gestionados, son internos al proyecto, como los cambios de requisitos o la falta de comunicación, dejando de lado los externos, que no se pueden manejar a nivel del equipo, tales como productos de terceros que no funcionan como se esperaba, factores externos de los que depende el proyecto, pérdida o corrupción de los datos en los sistemas del equipo. En nuestro caso, no parece apropiado utilizar una técnica para la gestión continua de los riesgos externos, esta consiste en la elaboración de una tabla en donde se identifica cada **riesgo**, con su probabilidad de ocurrencia, y el costo en caso de que ocurra. A partir de estos datos, se puede calcular lo que se conoce como exposición al **riesgo**, permitiendo la construcción de un “Risk Burn-Down Chart” que muestra la misma a lo largo de los sprints.

Como en la gestión de riesgos, es fundamental un responsable que lleve a cabo esta tarea, decidimos que, en nuestro caso, el mismo sea el SCRUM MASTER.

## Conclusión

La Gestión de Riesgos, es un proceso muy importante dentro del ámbito de las empresas de desarrollo de software, la mayoría de las entidades y empresas del rubro la tienen implementadas. Por lo que cualquier empresa que no estuviera aplicando esta gestión estaría perdiendo los beneficios de implementar esta nueva forma de gestión con la consiguiente dificultad de alcanzar sus objetivos.

Además que la implementación de la Gestión de Riesgos es de un bajo costo, ya que existen en la actualidad un sinnúmero de modelos a aplicar según la empresa y actividad, que van desde modelos generales aplicable a cualquier empresa hasta Software más sofisticados y dedicados a algunos riesgos específicos como es el caso de los riesgos del tipo TI (tecnológicos informáticos), lo que son de un costo mayor.

Para implementar la Gestión de Riesgos en una empresa no hay que olvidar que una de las etapas más importantes es la de crear conciencia y sensibilizar al personal en la cultura de los riesgos, para que sean ellos mismos quienes puedan controlar y evaluar sus procesos.

# Plan de gestión de la configuración

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	Plan de gestión de la configuración.
<b>Referencia:</b>	GeoP_Proyecto_PlanDeGestionConfiguracion.docx
<b>Autores:</b>	Ezequiel Bär Coch
<b>Fecha:</b>	29/05/2014

### Histórico de Versiones.

<b>Versión</b>	<b>Fecha</b>	<b>Estado</b>	<b>Responsable</b>	<b>Cambios</b>
1.0_DrafA	28/05/2014	Pendiente de revisión	Ezequiel Bär Coch	Creación del documento
1.0	03/06/2014	Aprobado	Leonel Romero [Revisor]	
1.1_DraftA	14/06/2014	Pendiente de Aprobación	Marcos Barrera [Actualizador]	Se actualiza información de versionado de documentos. Se actualiza el sistema de gestión de configuración (GitHub).
1.1	19/06/2014	Aprobado	Leonel Romero [Revisor]	
1.2_DraftA	03/02/2015	Pendiente de revisión	Ezequiel Bär Coch	Se actualiza y modifica la sección de documentos a ser dispuestos bajo el control de gestión de configuración. Se modifican detalles de redacción y fuentes.
1.2_DraftA	06/02/2015	Aprobado	Ignacio Frigerio	

## Introducción

### Propósito

El objetivo del Plan de Gestión de la Configuración (CM), es proporcionar una visión general de la identificación de los elementos de configuración (CI), control de cambios y las auditorías de configuración en un alto nivel para el desarrollo del producto GeoParking.

### Alcance

El plan de gestión de la configuración involucrará desde el Sprint 0 al Sprint 12 definidos en el **Estudio Inicial**. El documento permitirá mostrar los estándares de etiquetación de los productos de trabajo. Así mismo definirá el tipo de nomenclatura utilizada para el control de las versiones de los documentos que se encuentran dentro de los elementos de gestión de la configuración.

Este proceso de control de cambios se aplica a los productos de línea base creados o gestionados por los miembros del sistema, incluyendo:

- El software lanzado a producción o que se encuentra en versión beta.
- Manual de procedimientos.
- Manual de usuario web.
- Manual de usuario móvil.
- Documentación técnica de capacitación.
- Plan de gestión de riesgos.
- Plan de impacto ambiental.
- Plan de normas de desarrollo.
- Plan de testing.

### Acrónimos

CCB	Configuration Control Board (miembros del equipo GeoParking).
CM	Control Management
GCS	Gestión de la Configuración del Software
ECS	Elementos de la Configuración de Software
CMO	Configuration Management Officer
CI	Configuration Item

### Definiciones

Línea Base	Un documento o producto oficial aprobado que sirve como punto de partida para futuras versiones.
Configuration Control Board	Tiene por objetivo revisar y aprobar los cambios sugeridos a un producto.
Petición de Cambio	Una solicitud que alguien ha presentado al sistema de control de cambios que describe un problema de software, una mejora solicitada, una propuesta de cambio en los

	requisitos de un producto en fase de desarrollo, o un nuevo proyecto que se propone.
Stakeholder	Persona que directa o indirectamente se ve afectada por el sistema y que puede afectar el proyecto.
Configuration Item	Los elementos que son puestos bajo el control de gestión de la configuración.

## Organización

### Sistema de Gestión de la Configuración

GitHub, un sistema de control colaborativo de revisión y desarrollo de software, es una herramienta del tipo cliente, que se utiliza para almacenar todas las versiones del software y dar seguimiento de los cambios y líneas de base del proyecto. Esta estructura de configuración estará almacenada en el repositorio de GitHub al cual todos los miembros del equipo tienen acceso.

### Personal, Roles y Responsabilidades

#### CCB

La CCB, está integrada por el Scrum Master, Developer Team, Testing Team, CM y DBA que serán roles intercambiables a lo largo de los Sprints. Cualquier cambio en los requisitos deberá ser aprobado por el CCB. Las responsabilidades contemplarán:

- Revisar todas las peticiones de cambio y proporcionar los datos necesarios para determinar la disposición, tal como se describe en la Sección 3.4.
- Asignar las peticiones de cambio aprobadas, una fecha de implementación y un equipo.
- Asegurar que las acciones tomadas por las peticiones de cambio son de manera oportuna.

#### Configuration Management Officer (CMO)

Es el encargado de administrar el sistema de gestión de la configuración (Ezequiel Bär Coch), introducir las líneas base, otorgar permisos y administración de usuarios.

### Herramientas

Las siguientes herramientas son usadas para administrar las líneas base del proyecto:

- Para gestionar las historias de usuario, tenemos las ceremonias propias de SCRUM (planning, retrospective, review, daily, etc.), documentos y código fuente. La herramienta utilizada para la gestión de configuración es GitHub que almacenará las mismas en un repositorio en la nube propio de GitHub.

## Procedimientos

### Estimación de tiempo para identificación de Elementos

Con base a la planificación del proyecto, el CMO, determino que el tiempo estimado para la identificación de los elementos tomara un total de 2 semanas (es decir un Sprint) a partir del día 29 de Mayo.

### Identificar elementos de configuración

La identificación de los elementos de configuración (CI) consiste en determinar los componentes del sistema y documentos, que se pondrán bajo la identificación exclusiva de un identificador único, y hacerlas accesibles al equipo. Un esquema de correcta identificación, otorga a cada elemento trazabilidad entre el elemento y su información de estado.

Cada elemento se identificara de la siguiente manera

**[GeoP\_TipoDeDocumento\_NombreDelElemento\_Versión]**

Por ejemplo:

(GeoP\_Proyecto\_Estudiolnicial \_2.3.1)

Significa:

**Proyecto**                      *Geo Parking*

**Tipo de Documento**      *Proyecto*

**Nombre Del Elemento**    *Estudiolnicial*

**Versión del Elemento**    *2.3.1 (Segunda Versión. Tres modificaciones. Un error eliminado)*

**Nota:** En todos los casos, los nombres de los documentos serán las primeras letras capitales de los mismos. En el caso de coincidir, se deberá elegir alguno y agregarle la primera letra que siga en la nomenclatura del mismo.

Los tipos de documentos y sus acrónimos se muestran a continuación:

- NormaDesarrollo – Normas de Desarrollo
- PlanPrueba – Planes de pruebas
- GestionRisegos – Gestión de Riesgos
- Estudiolnicial – Estudio Inicial
- PlanGestionConfiguracion – Plan de Gestión de Configuración
- *ImpactoAmbiental – Impacto ambiental*
- *Otros documentos podrán ser agregados al ser identificados.*

A continuación se presentan los elementos que se pondrán bajo la gestión de la configuración:

- Product Backlog
- Sprint Backlog
- Sprint Reviews
- Sprint Retrospective

- Sprint Product Burndown Chart
- Product BurnUp Chart
- Manuales de Usuario (Web y móvil)
- Código fuente web.
- Código fuente móvil.
- Diagramas de Diseño (Interacción, Comunicación, Clases, Implementación, Interfaces, Estados)
- Diagramas Entidad Relación
- Otros

### **Establecer un sistema de administración de configuración**

Para seleccionar el sistema que servirá como gestor de la configuración, se tomaron en cuenta los siguientes puntos:

- Que la versión del software no sea de prueba o pago.
- Permita administrar a los usuarios que tendrán acceso al sistema.
- Permita otorgar permisos a los usuarios que accederán al sistema.
- Que sea un sistema fácil de usar.
- Que no sea un plugin de un ambiente de desarrollo (IDE).
- Que se pueda utilizar en distintos sistemas operativos.
- Que permita solucionar los conflictos (a la hora de crear merges) que surjan de una manera eficaz.

Sistema	Prueba o Paga	Usuarios	Permisos	Usabilidad	Conexión con SVN	Plugin	MultiOS	Resolución de Conflictos	Total
RiouxSVN	No	Hasta 5	Si	Si	No	Si	Si	Si	6
TortoiseHg	No	Hasta 5	Si	Si	Si	No	Si	Si	9
GitHub	No	Ilimitado	Si	Si	No	Si	Si	Si	9

**Tabla 1. Comparación de sistemas de gestión de la configuración**

Con respecto a la Tabla 1. Comparación de sistemas de gestión de la configuración, se puede ver que el sistema que se usara es GitHub, ya que es el más completo según el análisis que se realizó.



## Crear o liberar las líneas base

Para la creación o liberación de una línea base se debe seguir el proceso que se presenta en la **Ilustración 1. Proceso de liberación de líneas base**, la persona interesada en la creación o liberación de líneas base debe obtener la autorización de la CCB, cualquier integrante del equipo puede presentar una propuesta para creación o liberación de línea base pero esta solo puede ser autorizada por la CCB, esta propuesta debe de realizarse a través del formato correspondiente.

Si la petición de liberación fue aprobada por el CCB, el administrador de la Herramienta de Gestión de la Configuración deberá liberar los elementos de línea base que se le soliciten y deberá de registrar la salida de dichos elementos.

El CCB informará que el conjunto actual de líneas base esté disponible a los interesados.

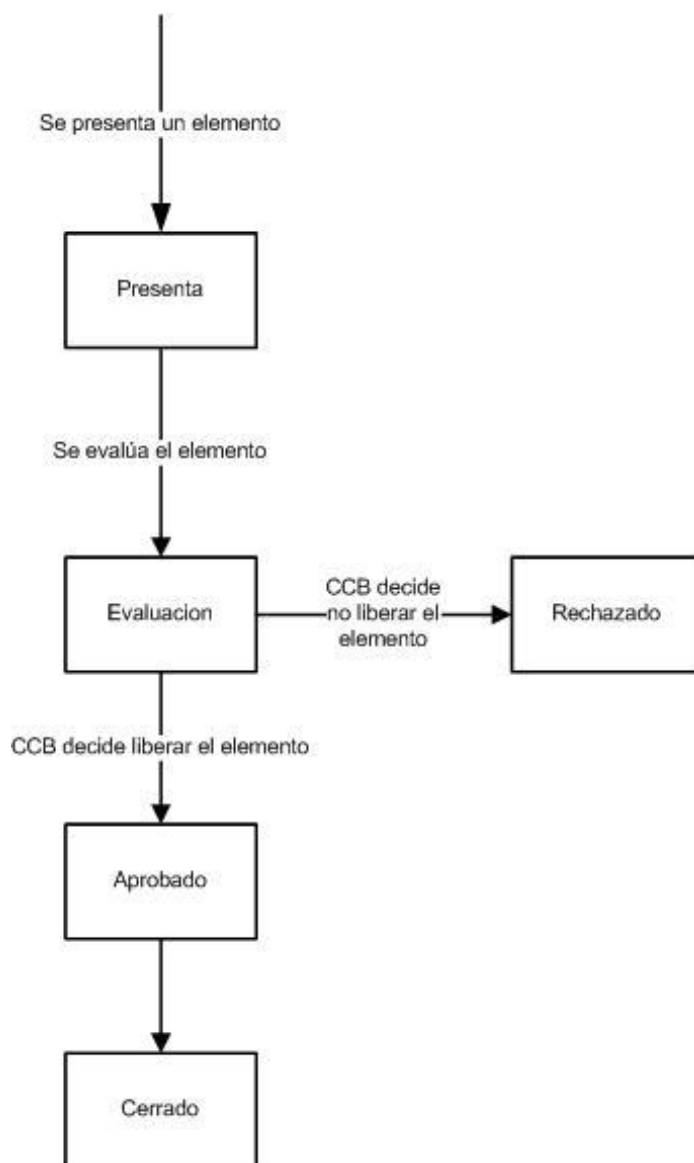


Ilustración 1. Proceso de liberación de líneas base

## Seguir las peticiones de cambio

Cualquier cambio que afecta a los requisitos de línea base debe ser presentado a la CCB como una petición de cambio. La **Ilustración 2. Proceso de petición de cambio**, muestra el diagrama de flujo típico de una solicitud de cambio. El CCB debe analizar y considerar cada petición de cambio. Si la solicitud es rechazada, puede ser redefinida y volver a presentarla. Si se acepta, los requisitos serán de nuevo línea base y todos los potencialmente afectados por el cambio serán informados. Esta petición debe de realizarse a través del formato correspondiente.

Cada petición de cambio presentada que sea aceptada se debe revisar o asignar una prioridad, y proporcionar una evaluación de impacto (una estimación aproximada del nivel de esfuerzo requerido para la ejecución, y el impacto de otras actividades actuales y previstas).

Se asigna una prioridad a cada petición de cambio cuando se recibe. La prioridad de una petición de cambio es asignado, ya sea por el autor o por el desarrollador líder. El desarrollador líder tiene la autoridad para modificar la prioridad de cualquier petición de cambio. Un cambio necesario tan pronto como sea posible, sin pasar por el calendario de lanzamientos regulares, se le asigna una prioridad urgente. Todos los cambios, independientemente de su prioridad, siguen el mismo proceso de aprobación.

Al recibir una petición de cambio, el CCB evalúa el cambio, en caso de ser necesario se pone en contacto con el autor del cambio, procesa la solicitud del cambio, y recomienda un calendario para la aplicación de los cambios aprobados.

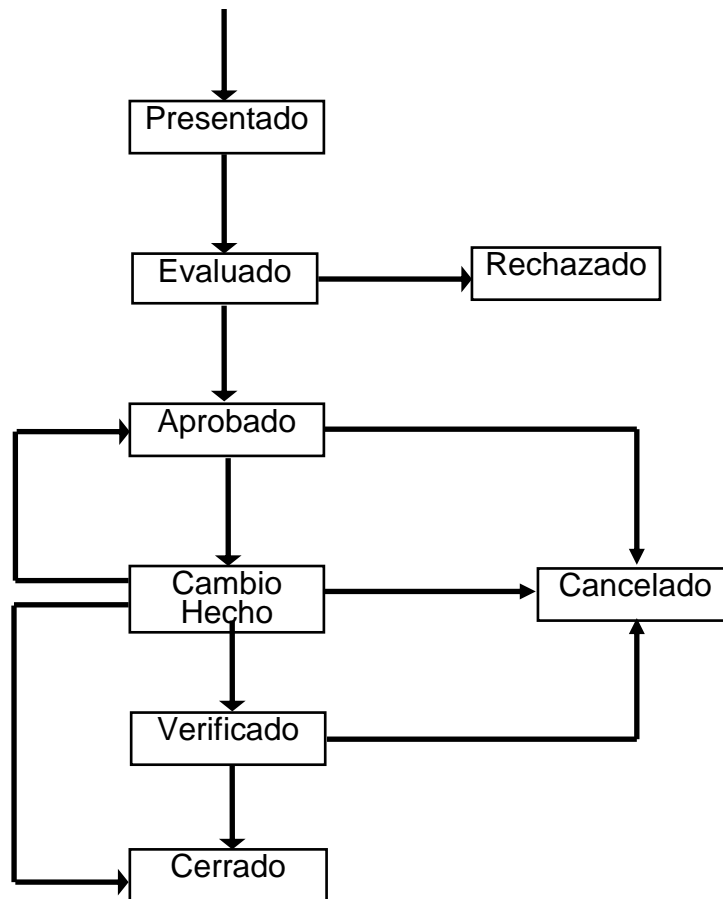


Ilustración 2. Proceso de petición de cambio

## Controlar los elementos de configuración

La CMO recoge los datos necesarios para producir informes útiles para el CCB, y el Scrum master del proyecto.

Para la gestión del cambio, la CMO recoge información de identificación correspondiente a cada petición de cambio recibida y su estado en la base de datos de peticiones de cambio, tal como se define en el procedimiento de peticiones de cambio. Después la CMO prepara sus respectivos informes.

Para el estado de los elementos de configuración, la CMO recoge información de identificación correspondiente a cada elemento de configuración controlada, es decir, la versión actual, el historial de revisiones y el subsistema asociado. Al final de cada entrega, los elementos de configuración se actualizan como se define en el procedimiento de identificación de elementos bajo configuración. La CMO prepara informes conforme a lo solicitado sobre el estado de CI, que detalla las nuevas solicitudes de cambio, las solicitudes de cambio recientemente aprobadas, y las solicitudes de cambio cerradas. Se preparan informes de liberación por la CMO para que se ingresen en los informes de la liberación.

## Realizar auditorías de configuración

Las auditorías de configuración consisten en revisiones donde se compara el proceso de CM o una configuración de un producto con los requisitos para determinar si estos se cumplen con lo definido en este documento.

Con base a lo anterior, se identifican las siguientes fechas para las auditorías de la gestión de la configuración:

- 20-08-2014: Corresponde al Sprint 5 según calendarización.
- 20-10-2014: Corresponde al Sprint 11 según calendarización.

## Auditorías de línea base

El Scrum Master de turno, trabajará con los miembros del equipo para llevar a cabo una auditoría a las líneas base por cada liberación. Esta auditoría de base verifica el contenido de la versión completa, y todos los cambios que deberían realizarse han sido verificados por el equipo. Las revisiones de configuración se realizarán periódicamente para verificar la exactitud de la información de los elementos de configuración. El objetivo de la revisión es para verificar que todos los componentes del sistema han sido correctamente identificados y que todos los cambios en él han sido debidamente administrados. Las revisiones periódicas de configuración también permitirán evaluar la eficacia del proceso de CM y para identificar posibles modificaciones. El CMO seguirá de cerca la configuración de los CI y trabajará con los involucrados para asegurar que los procedimientos de CM se utilizan adecuadamente antes de las liberaciones principales del software.

A continuación se presentan los tres diferentes tipos de auditorías a líneas base para llevar a cabo:

1. Las auditorías de líneas base:
  - Una auditoría de línea base se llevará a cabo por la CMO cuando cada línea base se halla establecido y se esté listo para pasar al siguiente sprint. Los resultados de esta auditoría se documentarán y se proveen al equipo de desarrollo de turno.
  - Las auditorías de construcción de código serán llevadas a cabo por la CMO cuando una línea base construida esté lista para avanzar a la fase de pruebas. Esta auditoría verificará el contenido de la construcción en comparación con el contenido previsto en la planeación. Los resultados de esta auditoría se documentarán y se suministra con el equipo de pruebas para que conozcan que la aplicación está lista para ser probada.
2. Cerca de la conclusión del proyecto, una auditoría funcional de configuración (FCA) se llevará a cabo por la CMO. La FCA valida que un producto de trabajo cumple con sus requisitos de rendimiento. La CMO analizará la:
  - Matriz de trazabilidad de Historias de Usuario
  - Hojas de certificación y evidencia objetiva de que cada requerimiento se validó.
3. Cerca de la conclusión del proyecto, una auditoría física de configuración (PCA) se llevará a cabo por la CMO. El PCA valida la coherencia entre el producto y su documentación técnica. La CMO analizará:
  - Lista de elementos a ser inspeccionados (repositorio)

Registro de estado (repositorio actualizado)

# Normas de Desarrollo

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	Normas de desarrollo
<b>Referencia:</b>	GeoP_Proyecto_NormasDesarrollo.docx
<b>Autores:</b>	Marcos Barrera
<b>Fecha:</b>	29/05/2014

### Histórico de Versiones.

<b>Versión</b>	<b>Fecha</b>	<b>Estado</b>	<b>Responsable</b>	<b>Cambios</b>
1.0_DraftA	29/05/2014	Pendiente de Revisión	Marcos Barrera [Autor]	
1.0_DraftB	31/05/2014	Pendiente de Revisión	Leonel Romero [Revisor]	
1.0	17/09/2014	Aprobado	Marcos Barrera	

## Introducción

El uso de un estándar de programación o codificación estándar, es de gran relevancia, al momento de desarrollar un producto de software, lo cual repercute en la calidad de éste. El uso de un estándar de programación permite mejorar la práctica, elusión de problemas (ambigüedad) en el código, logrando un conocimiento y un mejor seguimiento para el equipo de desarrollo. En este documento se da a conocer un estándar de codificación orientado principalmente al lenguaje de programación C#.

Este documento será utilizado por todos los miembros del equipo tanto para desarrollo como testing del mismo.

## Comentarios

### Recomendaciones generales

Los comentarios dan información sobre lo que hace el código en el caso que no sea fácil comprenderlo con una lectura rápida. Se usan para **añadir información** o **para aclarar secciones de código**. Los comentarios se pueden escribir en diferentes estilos dependiendo de su longitud y su propósito. En cualquier caso seguiremos las siguientes **reglas generales**:

- Los **comentarios** en general se escriben **en líneas que no contienen código** y antes del código que queremos clarificar. Esta regla se aplica siempre si el comentario tiene más de una línea.
- **Sólo** en dos casos se permite **poner en la misma línea** un comentario y una instrucción: **comentarios a una definición de variable**, que explica la finalidad de esta variable, y un **comentario** para indicar **final de una estructura del lenguaje**.

#### 1.1. Documentación XML. Etiquetas recomendadas

La documentación debe estar en XML bien formado. Si el XML no está bien formado, se generará un error de advertencia y el archivo de documentación incluirá un comentario que mencione el error encontrado.

Etiquetas básicas recomendadas:

**<summary>**: se utiliza para describir brevemente un tipo o un miembro de tipo. El texto de esta etiqueta es la única fuente de información sobre el tipo en *IntelliSense*.

**<summary>Descripción</summary>**

**<remarks>**: se utiliza para agregar información sobre un tipo, de modo que completa la información especificada con **<summary>**. **<remarks>Descripción detallada</remarks>**

**<param>**: se debe utilizar en el comentario de una declaración de método para describir uno de los parámetros del método.

**<param name='NombreParámetro'>Descripción</param>**

**<returns>**: se debe utilizar en el comentario de una declaración de método para describir el valor devuelto.

**<returns>Descripción</returns>**

**<paramref>**: proporciona un modo de indicar que una palabra es un parámetro. El archivo XML se puede procesar de manera que aplique formato a este parámetro de algún modo diferente.

**<paramref name="NombreParámetro"/>**

**<exception>**: permite especificar las excepciones que se pueden iniciar. Esta etiqueta se aplica a una definición de método.

**<exception cref="member">Descripción</exception>**

**<value>**: permite describir una propiedad. Al agregar una propiedad a través del asistente para código se creará una etiqueta **<summary>** para la nueva propiedad. A continuación, se debe agregar manualmente una etiqueta **<value>** que describa el valor que representa la propiedad.

**<value>Descripción del valor de la propiedad</value>**

## Tipos de Comentarios

Con todo esto se pueden definir los siguientes tipos de comentarios:

**Comentarios de clase:** Usados en la definición de una clase. Opcionalmente puede contener información acerca de la versión.

### Ejemplo:

**///  
///** **<summary>**Descripción genérica de la clase**</summary>**

**///  
///** **<remarks>**Descripción detallada de la clase**</remarks>**

**Comentarios de variables:** En el caso de que una variable requiera comentario éste deberá aparecer al estilo XML. Si únicamente son aclaraciones, se realizarán en la misma línea en que se declara la variable y serán muy breves.

**Ejemplo:** **///  
///** **<summary>**Descripción breve de la variable**</summary>** private int iNumPruebas; **//**  
Contador pruebas realizadas

**Comentarios de métodos:** Usados en la definición de los métodos, simplemente describen su funcionalidad.

### Ejemplo:

```

/// <summary>Descripción breve del método</summary>
/// <remarks>Descripción detallada del método</remarks>
/// <param name="param1">Descripción parámetro 1</param>
/// <returns>Información valor de retorno</returns>
/// <exception cref="ex1">Descripción excep1</exception>

```

**Comentarios cortos:** Se usan para describir funcionalidades o datos. Deben situarse en la línea inmediatamente anterior a la funcionalidad que se va a describir. También se usan para indicar el final de una estructura. En caso de utilizar este tipo de comentario, seguir las siguientes reglas:

- Utilizar uno o más tabuladores para separar la instrucción y el comentario.
- Si aparece más de un comentario en un bloque de código o bloque de datos, todos comienzan y terminan a la misma altura de tabulación.

**Ejemplo:**

```

// Obtenemos el menor de los datos
for (...)
{
...
} // Fin del FOR

```

## Otras recomendaciones

Otras recomendaciones sobre el uso de comentarios son:

- Cuando modifique el código, mantenga siempre actualizados los comentarios circundantes.
- Después de la implementación, quite todos los comentarios temporales o innecesarios para evitar cualquier confusión en la fase de mantenimiento.
- Use los comentarios para explicar el propósito del código. No los use como si fueran traducciones interlineales.
- Realice los comentarios con un estilo uniforme en toda la aplicación.



## Sangrías, espacios y líneas en blanco

Los espacios en blanco facilitan la lectura y el mantenimiento de los programas. Los espacios en blanco que se pueden utilizar son: las sangrías, los caracteres espacio y las líneas en blanco.

### 1.2. Sangrías

El sangrado se utiliza para mostrar la estructura lógica del código. El sangrado óptimo es el formado por **cuatro espacios**. Es un compromiso entre una estructuración legible y la posibilidad de que alguna línea (con varios sangrados) del código supere el ancho de una línea de una hoja de papel o del monitor.

El sangrado se debe realizar con tabulaciones, por tanto se debe fijar éste en cuatro caracteres. Es la opción por defecto del entorno Visual Studio .NET.

### 1.3. Caracteres espacio

Los espacios en blanco sirven para facilitar la lectura de los elementos que forman una expresión. Los espacios en blanco se utilizan en los casos siguientes:

- Las variables y los operadores de una expresión deben estar separados por un elemento en blanco.

*Ejemplo: Espaciado de operadores*

```
fMedia = iSuma / iCuenta;
```

- Las listas de definición de variables y las listas de parámetros de una función se debe separar por un espacio en blanco.

*Ejemplo: Lista de parámetros*

```
AbrirBusqueda (iTarjeta, iMovil);
```

### 1.4. Líneas en blanco

Se utilizan para separar “párrafos” o secciones del código. Cuando leemos un programa entendemos que un fragmento de código entre dos líneas en blanco forma un conjunto con una cierta relación lógica

Veamos como separar secciones o párrafos en un programa:

- Las secciones que forman un programa se separan con al menos una línea en blanco (declaración de constantes, declaración de variables, métodos,...).
- Dentro de un subprograma se separan con una línea en blanco los fragmentos de instrucciones muy relacionadas entre sí (por ejemplo, conjunto de instrucciones que realizan una operación).

### 1.5. Uso de las llaves

La forma que tiene C# de agrupar instrucciones en bloques es utilizar las llaves {}. Su colocación se debe hacer en líneas reservadas para cada una de ellas, sin ninguna otra instrucción en la línea. Ambas deben ir en la misma columna que la instrucción que la precede.

*Ejemplo:*

```
for (iContador = 0; iContador < iNumIteraciones; iContador++)  
{  
...  
}
```

Se permiten algunas excepciones a esta norma en las sentencias condicionales y de repetición donde la llave de apertura puede estar al final de la sentencia de control o donde tras la llave de cierre se permiten sentencias condicionales.

**Ejemplo:** llave al final de la sentencia

```
if (iContador == 0)
{
...
}
else {
...
}
```

**Ejemplo:** sentencias tras la llave de cierre

```
do
{
...
} while (iEstado != iResultOk)
```

Se debe evitar en la medida de lo posible omitir las llaves, incluso en aquellos bloques que encierran una única sentencia.

**Ejemplo:** llaves en bloques con sentencias únicas

```
if (iContador == 0)
{
iResultado = RET_OK;
}
```

### 1.6. Código sin finalizar

Siempre que partes de la funcionalidad no se implementen, queden inacabadas o sea previsible concluir las en un futuro, es conveniente documentarlo de modo que sea fácilmente localizable. Con esta finalidad, el entorno de desarrollo de C# mantiene un listado de tareas pendientes que se pueden obtener directamente de los comentarios de código. Así, cuando se deba completar cualquier tipo de código se documentará con “//TODO:” más la descripción de la funcionalidad pendiente, y ésta aparecerá en el editor de tareas cuando se muestren todas o las de comentario.

**Ejemplo:** documentación de código sin finalizar

```
private void CalcularBalance ()
{
// TODO: Calcular el Balance de la cuenta
}
```

### 1.7. Conversión de tipos

Cuando sea necesario convertir datos de un tipo a otro, es aconsejable que éstas se hagan de modo explícito, aunque muchas de ellas las realice el compilador automáticamente (implícito).

**Ejemplo:** conversión explícita

```
long lValor = 3000;
int iValor = (int) lValor;
```

## CONVENCIÓN DE NOMBRES

Los identificadores que dan nombre a la aplicación, clases, métodos, variables, tipos, etc. han de colaborar en la autodocumentación del programa, aportando información sobre el cometido que llevan a cabo (*el qué y no el cómo*). Para elegir los nombres se deben seguir las siguientes recomendaciones generales

- Elegir nombres comprensibles y en relación con la tarea que corresponda al objeto nombrado.
- Utilice palabras completas siempre que sea posible. Las abreviaturas pueden adoptar muchas formas y en consecuencia, pueden resultar confusas. En el caso en que sea estrictamente necesario usarlas, elija las abreviaturas adecuadamente.
- Cuando se trabaje con siglas o acrónimos, éstos deben mantener todos sus caracteres es mayúsculas.

**Ejemplo:** *Uso siglas o acrónimos*

*int GenerarHTML (...)*

- Utilizar prefijos definidos para cada uno de los elementos.

### 1.8. Variables y objetos

La notación húngara es un sistema usado normalmente en la comunidad de desarrolladores para crear nombres cuando se programa en Windows. Consiste en prefijos en minúsculas que se añaden a los nombres de las variables y que indican su tipo. El resto del nombre indica, lo más claramente posible, la finalidad del elemento.

<b><i>Prefijo</i></b>	<b><i>Tipo y Significado</i></b>	<b><i>Ejemplos</i></b>
<b><i>b</i></b>	Booleano ( <b>bool</b> )	bool bNombre;
<b><i>ch</i></b>	Carácter ( <b>char</b> )	char chNombre;
<b><i>s</i></b>	Cadena de texto ( <b>string</b> )	string sNombre;
<b><i>bt</i></b>	1 byte ( <b>byte</b> )	BYTE btNombre;
<b><i>i</i></b>	Entero con signo ( <b>int</b> )	int iNombre;
<b><i>sh</i></b>	(short)	short shNombre;
<b><i>l</i></b>	Entero largo ( <b>long</b> )	long lNombre;
<b><i>db</i></b>	(double)	double dbNombre;
<b><i>f</i></b>	(float)	float fNombre;
<b><i>d</i></b>	(decimal)	decimal dNombre ;

<b>dt</b>	(DateTime)	DateTime dtNombre;
<b>st</b>	Estructuras ( <b>struct</b> )	struct stNombre {};
<b>e</b>	Enumeraciones (enum)	enum eNombre {};
<b>o</b>	Objeto o instancia de una clase	Raiz oRaiz;
<b>ds</b>	(DataSet)	DataSet dsNombre;
<b>dt</b>	(DataTable)	DataTable dtNombre;
<b>dr</b>	(DataRow)	DataRow drNombre;
<b>dc</b>	(DataColum)	DataColum dcNombre;
<b>dv</b>	(DataView)	DataView dvNombre;
<b>drd</b>	(DataReader)	DataReader drdNombre;
<b>ht</b>	(Hashtable)	Hashtable htNombre;

### 2.2.2. Formularios e Interfaces

<b><i>Prefijo</i></b>	<b><i>Tipo y Significado</i></b>	<b><i>Ejemplos</i></b>
<b>I</b>	Nombres de interfaces	interface INombreInterfaz
<b>Form</b>	Formulario	class FNombreFormulario

### 1.9. Controles Windows

Del mismo modo que con las variables y objetos se notarán del siguiente modo:

<b><i>Prefijo</i></b>	<b><i>Tipo y Significado</i></b>	<b><i>Ejemplos</i></b>
<b>lbl</b>	Label	lblNombre
<b>lnk</b>	LinkLabel	lnkNombre
<b>btn</b>	Button	btnNombre

<b>txt</b>	TextBox	txtNombre
<b>mnm</b>	MainMenu	mnmNombre
<b>chk</b>	CheckBox	chkNombre
<b>rdb</b>	RadioButton	rdbNombre
<b>grp</b>	GroupBox	grpNombre
<b>pct</b>	PictureBox	pctNombre
<b>pnl</b>	Panel	pnlNombre
<b>dg</b>	DataGrid	dgNombre
<b>lsb</b>	ListBox	lsbNombre
<b>cmb</b>	ComboBox	cmbNombre
<b>lsv</b>	ListView	lsvNombre
<b>trv</b>	TreeView	trvNombre

### 1.10. Excepciones

Para las propiedades y las clases genéricas no se aplican estas reglas:

<b><i>Tipo y Significado</i></b>	<b><i>Ejemplos</i></b>
Propiedades	string Nombre;
Nombres de clases genéricas	class CNombreClase

## GUÍA DE ESTILO

### 1.11. Ficheros

Por norma general un fichero debe contener una **única clase**. Sólo se incluirán más cuando sea estrictamente necesario.

Las **líneas de código** no deben exceder de los **80 caracteres**, incluyendo las tabulaciones y los comentarios en línea. Las líneas largas deben ser separadas en varios renglones, de modo que al imprimir el código sea legible. Cuando esto ocurra la segunda línea y las posteriores se deben indentar según las circunstancias de cada línea.

**Ejemplo:** Cadena de texto larga

```
MessageBox.Show("Esto es un ejemplo de línea larga que se debe indentar en la segunda línea");
```

**Ejemplo:** Sentencia condicional

```
if ( ( iEstado == Estado.A) && (iFase == Fase.F1) ) ||  
( iEstado == Estado.B) && (iFase == Fase.F2) ) )  
{  
}
```

Otro aspecto importante es la indentación del código. Esto depende de la configuración de los distintos editores que se usen. En la medida de lo posible se debe indentar mediante **tabuladores** de **4 caracteres**, como se define en el apartado de "Sangría".

La estructura genérica de los ficheros será

- Sentencias *using*
- Definición del *namespace*
- Declaración de la clase
  - ✓ Variables o campos
  - ✓ Delegados y eventos
  - ✓ Propiedades
  - ✓ Métodos

La organización interna de los campos y los métodos en función de su ámbito de acceso será:

- Públicos
- Protegidos
- Privados

### 1.12. Sentencias condicionales y de iteración

#### 1.12.2. Sentencia IF

A la sección *if...else* van siempre en una línea sangrada. Aunque está permitido, no se recomienda omitir las llaves cuando se englobe una única sentencia.

**Ejemplo:** sin llaves (no recomendado)

```
if (iValor1 < iValor2)  
iValor1=0;  
else
```

```
iValor1=1;
```

**Ejemplo:** con llaves

```
if (iValor1 < iValor2)  
{  
iValor1=0;  
}  
else {
```

```
iValor1=1;
}
```

Los *if* anidados deben seguir una secuencia de sangrado lógica.

**Ejemplo:** con llaves

```
if (iValor1 < iValor2)
{
iValor1=0;
}
else if (iValor1 < iValor3)
{
iValor1=1;
}
else {
iValor=2;
}
```

### 1.12.3. Sentencia SWITCH

Los valores asociados a la sentencia *switch* irán en línea aparte sangrada. El bloque de sentencias asociado comenzará en otra línea aparte y también sangrada. Todos los bloques de sentencias pertenecientes al *case* comenzarán en la misma columna. Cada bloque *case* se deberá separar del siguiente con una línea en blanco cuando englobe muchas líneas de código.

**Ejemplo:** Uso de switch

```
switch (iValor)
{
case Valor.Uno:
sTexto = "Valor A";
...
break;
case Valor.Dos:
sTexto = "Valor B";
...
break;
default:
sTexto = "Valor desconocido";
}
```

Cuando la discriminación se realice sobre tipos numéricos será obligatorio encapsular todos los valores posibles en *enumeraciones* que se autocomenten.

**Ejemplo:** Uso de case

```
switch (iValor)
{
case Valor.Uno: /// Nunca case 1: ¡!!!
sTexto = "Valor A";
...
break;
...
}
```

Es recomendable comentar el caso en que una sentencia *case* caiga dentro de otra, de modo que las dos ejecuten el mismo código

**Ejemplo:** *Uso de case con mismo código*

```
switch (iValor)
{
case Valor.Uno:: ///Valores impares case Valor.Tres:
sTexto = "Valor impar";
break;
...
}
```

Del mismo modo, cuando una sentencia *case* no tenga código asociado por el motivo que sea, se debe comentar este motivo.

**Ejemplo:** *Uso de case sin código*

```
switch (iTipo)
{
case Tipo.Redondo:
///Futura ampliación – sin implementar
break;
case Tipo.Cuadrado:
...
break;
...
}
```

#### **1.12.4. Sentencias de iteración**

Las sentencias pertenecientes a estructuras de iteración o repetición (*for*, *while*, *do...while* y *foreach*) van siempre en una nueva sangrada y con su código indentado. Del mismo modo que las sentencias condicionales, se debe evitar la omisión de las llaves.

**Ejemplo:** *for sin llaves*

```
for (iContador=0;... ;...)
iSuma += iTipo;
```

**Ejemplo:** *for con llaves*

```
for (iContador=0;... ;...)
{
iSuma += iTipo;
...
}
```

#### **1.13. Variables y objetos**



Todas las variables deberán tener nombres significativos. Los nombres que formen una variable comenzarán en mayúsculas y no tendrán ningún carácter de separación (notación Pascal). Al nombre de la variable se le debe anteponer siempre y sin excepción alguna su tipo según el convenio de nombres.

**Ejemplo:** variables correctas

```
int iValor;  
string sTextoError;  
CCuadrado oCuadrado;
```

**Ejemplo:** variables incorrectas

```
int Valor;  
string sTextoeerror;  
string stextoeerror;
```

Las variables deben declararse siempre al principio de las clases o métodos, una por línea, con comentarios si se estima oportuno e inicializándolas según convenga

**Ejemplo:** declaraciones correctas

```
private float CalcularRaizCuadrada (int iValor)  
{  
    // Variables auxiliares  
    float fResultado;  
    string sTexto = "";  
    bool bExacto = false;  
    CRaiz oRaiz;  
    ...  
}
```

Los nombres de variables booleanas deben ser lo suficientemente claros como para expresar el valor que encierran ante una pregunta.

**Ejemplo:** variables booleanas

```
private bool bFinalizado;  
if (bFinalizado)  
{  
}
```

En la medida de lo posible se deben evitar las variables públicas y convertirlas en privadas o protegidas con su propiedad de acceso a ellas.

#### **1.14. Constantes**

Todas las cantidades que permanezcan constantes deben nombrarse con *const*. Este tipo de constantes respetarán las normas comunes de las variables.

**Ejemplo:** uso de constantes

```
public const int iValor = 10;
```

#### **1.15. Propiedades**

Para la notación de una propiedad se usará el mismo nombre que la variable sobre la que actúa sin la nomenclatura del tipo.

**Ejemplo:** definición de atributos de clases

```
class CPoligono  
{  
    private int iNumLados;  
    public int NumLados  
    {  
        get  
        {  
            return iNumLados;  
        }  
    }  
}
```

```
set
{
iNumlados = value;
}
}
}
```

### 1.16. Operadores

Cuando se usen operadores, principalmente cuando tomen dos parámetros, se deberá dejar un espacio antes y otro después del operador.

**Ejemplo:** uso de operadores

```
iNumPruebas += 5;
```

Es recomendable el uso de paréntesis para eliminar las ambigüedades que puedan surgir por desconocimiento de las prioridades de los operadores.

**Ejemplo:** uso de paréntesis es operadores

```
iResultado = (iDato1 * iDato2) / iDato3
```

### 1.17. Regiones

La sentencia **#region** permite especificar un bloque de código que se puede expandir o contraer cuando se utiliza la característica de esquematización del editor de código de Visual Studio.

Dentro de un fichero se deben especificar las siguientes regiones:

**Clase:** Como ya se ha comentado, es conveniente que exista una única clase por fichero. En el caso de que esto no se pueda cumplir hay que definir una región por cada clase.

**Ejemplo:** región a nivel de clase

```
#region Clase Balance
```

```
...
```

```
#endregion
```

- **Variables:** Debe englobar todas las variables definidas

**Ejemplo:** región a nivel de variables

```
#region Variables
```

```
...
```

```
#endregion
```

- **Delegados y Eventos:** Si existen delegados y/o eventos se englobarán dentro de una misma región.

**Ejemplo:** región a nivel de delegados y/o eventos

```
#region Delegados y Eventos
```

```
...
```

```
#endregion
```

- **Propiedades:** Sobre las propiedades que existan.

**Ejemplo:** región a nivel de propiedades

```
#region Propiedades
```

...

#endregion

- **Constructores:** Si se han definido constructores personalizados

***Ejemplo:** región a nivel de constructores*

#region Constructores

...

#endregion

- **Métodos:** Se englobarán todos los métodos

***Ejemplo:** región a nivel de métodos*

#region Métodos

...

#endregion

Si alguno de los elementos no se encuentran definidos, se debe eliminar la región correspondiente.

### **1.18. Métodos**

En los métodos se debe seguir el estándar **verbo-sustantivo** para asignar los nombres, donde el verbo ha de estar en infinitivo. Se deberán utilizar al menos dos palabras según el método **Pascal**. Además debe ser lo suficientemente significativo como para describir la acción que realiza.

***Ejemplo:** uso del estándar verbo-sustantivo*

*public int CalcularEdad ()*

*private int CalcularNomina ()*

Hay que evitar usar nombres imprecisos o que permitan interpretaciones subjetivas

***Ejemplo:** nombre impreciso o ambiguo*

*public int AnalizarEsto ()*

Si la lista de parámetros no cabe en una única línea, las siguientes deben estar indentadas hasta la posición en que comienza la lista de parámetros en la primera línea

***Ejemplo:***

*private void CalcularReducciones (int iTipo, bool bCeuta,  
 string sTexto)*

*{*

*...*

*}*

En los métodos que devuelvan valores debe haber una única sentencia *return* al final de la misma, con un valor que se haya ido rellenando en el cuerpo del método. Incluso si no se devuelve ningún valor, es una buena práctica introducir al final un *return* sin datos

**Ejemplo:** uso de return

```
bool GetTipo ()
{
    int iTipo;
    ... iTipo = Tipo.A;
    ... iTipo = Tipo.B;
    return iTipo;
}
```

Cuando se realicen sobrecargas de métodos, todos deben llevar a cabo una función similar.

### **1.19. Clases**

Los nombres de las clases deben seguir los conceptos básicos de las funciones, anteponiendo al nombre el prefijo correspondiente definido en el convenio de nombres de clases e interfaces

**Ejemplo:** nombres de clases

```
class RaizCuadrada
{
    ...
}
```

Como ya se ha comentado, en la medida de lo posible se debe respetar el que haya una única clase por archivo, cuyo nombre será el propio de la clase.

**Ejemplo:** nombres de ficheros

```
RaizCuadrada.cs
class RaizCuadrada
{
    ...
}
```

### **1.20. Interfaces**

Para la creación de interfaces se deben aplicar las mismas normas descritas para las clases, aplicando su prefijo.

**Ejemplo:** definición de interfaz

```
interface INuevoInterfaz
{
    ...
}
```

## APÉNDICE A: NOMENCLATURA DE ESPACIOS DE NOMBRES

Es particularmente importante diseñar correctamente los nombres de los espacios de nombres, para evitar el riesgo de hacer uso de un espacio de nombres idéntico al que utiliza otra persona.

Si dos espacios de nombres con el mismo nombre se instalan en el mismo equipo, se producirá un conflicto de espacios. Microsoft recomienda utilizar nombres de espacios que utilicen el esquema su <NombreEmpresa>.<Tecnología>.

## APÉNDICE B: EXCEPCIONES

Para el manejo de las excepciones se debe emplear el mecanismo propio del lenguaje: try -> catch -> finally.

Es conveniente seguir un orden lógico y homogéneo en la recogida de los errores. De este modo, las excepciones en el bloque *catch* se capturarán en este orden:

- Excepciones de operación
- Excepciones específicas
- Excepciones personalizadas
- Excepciones genéricas

Con el fin de homogeneizar los diferentes mensajes de cara a una estandarización de todos ellos, se generará un documento XML en el que se especificarán los mensajes de error que se pueden producir para un tratamiento más genérico de la información capturada dentro de las excepciones.

## APÉNDICE C: COMPATIBILIDAD CON OTROS LENGUAJES .NET

Para hacer que nuestro código sea compatible con otros lenguajes .NET se deben tener en cuenta los siguientes puntos.

### C.1. USO DE MAYÚSCULAS Y MINÚSCULAS

Se debe tener cuidado con la distinción entre mayúsculas y minúsculas. C# sí las diferencia, por lo que es perfectamente válido que dos nombres se diferencien únicamente porque algunas letras del primer nombre estén en mayúsculas, mientras que las correspondientes del segundo están en minúsculas. Esto cobra importancia cuando los ensamblados son llamados desde otros lenguajes .NET, como Visual Basic, que no distingue entre mayúsculas y minúsculas.

De este modo, aunque no es aconsejable, si utiliza nombres que se diferencian únicamente por el uso de mayúsculas y minúsculas, es importante hacerlo sólo en situaciones en las que ambos nombres no sean visibles fuera de la unidad del ensamblado. De lo contrario, impedirá que un código escrito en VB .NET sea capaz de acceder correctamente a su unidad.

### C.2. NOMBRES Y PALABRAS RESERVADAS

Es importante que los nombres no coincidan con palabras reservadas. Si coincide con las C#, con total seguridad obtendrá un error de sintaxis. Sin embargo, dado que existe la posibilidad de que sus clases sean utilizadas desde código escrito en otros lenguajes, es importante no utilizar palabras reservadas de éstos.

En general las palabras reservadas de C++ son muy similares a las palabras reservadas de C#, por lo que una confusión entre ellas es poco probable.

Por otro lado, es más probable tener problemas con VB .NET, que tiene muchas más palabras reservadas que C#, y no podemos confiar en el estilo de nombres Pascal debido a que VB .NET no distingue entre mayúsculas y minúsculas.

# Plan de testing

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	Plan de testing
<b>Referencia:</b>	GeoP_Proyecto_PlanTesting.docx
<b>Autores:</b>	Ignacio Frigerio
<b>Fecha:</b>	29/05/2014

### Histórico de Versiones.

<b>Versión</b>	<b>Fecha</b>	<b>Estado</b>	<b>Responsable</b>	<b>Cambios</b>
1.0_DraftA	29/05/2014	Pendiente de Revisión	Ignacio Frigerio [autor]	Se crea un esbozo del plan de testing
1.0_DraftB	23/08/2014	Pendiente de revisión	Leonel Romero	Se agrega contenido
1.0	05/10/2014	Aprobado	Ignacio Frigerio	

## Introducción

En el siguiente documento se detallara la modalidad con la cual se desarrollara el testing en el proyecto GeoParking, desde su definición hasta su ejecución, así como también los distintos tipos de tests a utilizar, reglas de nombrado, y demás detalles necesarios para realizar los diferentes tests.

## Plan de Testing

Para la realización del testing, nos valdremos de los siguientes elementos y actividades:

- Casos de Prueba
- Ciclos de Prueba Manuales
- Ciclos de Pruebas Automáticos
- Tests Exploratorios
- Tests de Sistema
- Tests de Regresión
- Casos de Prueba

Para realizar los tests nos valdremos de los casos de prueba, estos casos de prueba serán creados por el o los integrantes del equipo que en el sprint este cumpliendo el rol de tester. El caso de prueba contendrá un número secuencial identificador del mismo (Id), un Nombre, una breve descripción del escenario que se está probando, las precondiciones que se deben dar para poder ejecutar el test, una secuencia de pasos necesarios para ejecutarlo y el número de historia a la cual está asociado. Al momento de definir los pasos, se definirá el número identificador del paso, la acción que se realiza en el paso y el valor esperado al realizarlo; para la realización de los casos de prueba está definida la plantilla GeoP\_PlanPrueba\_TestCase-Template en el directorio \GeoParking\Proyecto\Templates.

Una vez se tiene toda la información necesaria, el caso de prueba pasara a un estado pendiente de revisión, en el cual otro integrante del equipo lo revisara y como resultado de la revisión se definirá si el caso de prueba es o no correcto.

Cabe aclarar que la definición de los casos de prueba se hará en paralelo a la codificación de la historia, por lo que no es necesario que este realizado el código para iniciar con las definiciones de los casos de prueba.

## Ciclos de Prueba Manuales

Los ciclos de prueba son ejecuciones de un conjunto de casos de prueba, realizadas manualmente. Estos ciclos se ejecutaran una vez finalizado el desarrollo del código, para corroborar su correcto funcionamiento y detectar posibles errores. Cada ejecución de un caso de prueba será como esta detallada en el caso de prueba correspondiente y se registrara su resultado en la plantilla correspondiente, la cual está ubicada en GeoP\_PlanPrueba\_TestCaseExec-Template en el directorio \GeoParking\Proyecto\Templates.



La ejecución de los casos de prueba debe registrar el resultado obtenido en cada paso, un estado el cual es Pasó en caso de que el caso de prueba se ejecute y se obtengan los resultados esperados o Fallo en caso contrario. En caso de que el caso de prueba contenga errores, se registrara la severidad de los errores (Mayor, Menor o Cosmético).

Un ciclo de prueba deberá ejecutar todos los casos de prueba asociados a las historias cuyo desarrollo se ha finalizado.

## Ciclos de Pruebas Automáticos

Los ciclos de prueba automáticos serán generados con el plugin Selenium para Mozilla Firefox, dicho plugin graba todos los pasos necesario de un caso de prueba, así como los valores esperados para la ejecución automática del mismo, y en la ejecución da como resultado si paso o fallo el caso de prueba, pudiendo verse el motivo del fallo.

Se ha decidido realizar los casos de prueba que contienen mayor cantidad de pasos de esta manera para poder replicarlos todas las veces necesarias, y si es posible también se automatizaran otros casos de prueba.

Para la automatización de los distintos casos de prueba se deberá desarrollar un script que cree las precondiciones en la base de datos y luego elimine los datos, los distintos datos que se requieran para las ejecuciones serán nombrados de la siguiente manera (siempre que no sea necesario un nombre específico, por ejemplo una dirección para ubicar en un mapa):

Tipo de dato	Formato
Cadena de texto	CPA_ + Nombre del dato a ingresar
Número	g*
Tipo predefinido (selección de un combo)	Según definido en el caso de prueba

\*se repite el digito la cantidad necesaria de veces para cumplir con el campo correspondiente

## Tests Exploratorios

Se realizaran tests exploratorios de al menos un escenario por historia de usuario que tenga 3 o más puntos de historia asociados.

Para su ejecución se seguirá la plantilla GeoP\_Exploratory-Template ubicada en \GeoParking\Proyecto\Templates

Donde se define el nombre, el propósito, el escenario y los resultados obtenidos.

## **Tests de Sistema**

Se ejecutaran todos los casos de prueba asociados a una historia en particular.

## **Tests de Regresión**

Antes de cada release se realizara un test de regresión, el mismo involucra todos los ciclos de prueba, tanto manuales como automáticos, de las historias incluidas en el release, y su propósito es asegurar el correcto funcionamiento de las funcionalidades incorporadas en el release.

# Estudio de Impacto Ambiental

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	Estudio de impacto ambiental
<b>Referencia:</b>	GeoP_Proyecto_ImpactoAmbiental.docx
<b>Autores:</b>	Lucas Toneatto
<b>Fecha:</b>	11/10/2014

### Histórico de Versiones.

<b>Versión</b>	<b>Fecha</b>	<b>Estado</b>	<b>Responsable</b>	<b>Cambios</b>
1.0_DraftA	11/10/2014	Pendiente de Revisión	Lucas Toneatto[autor]	
1.0	30/10/2014	Aprobado	Lucas Toneatto	

## Proyecto GeoParking

Una problemática que nos afecta desde hace algunos años y que se encuentra en continuo crecimiento es el tránsito en las grandes ciudades, que tiene su concepción en la creciente industria automotriz. Como consecuencia de la misma, observamos una gran dificultad por parte de los conductores para encontrar un lugar seguro donde dejar su vehículo mientras realiza sus actividades cotidianas o se encuentra haciendo turismo.

La oportunidad que identificamos es la realización de una aplicación web y móvil que les brinde a los conductores la mejor forma de llegar a un espacio con ciertas características deseadas, asegurando disponibilidad de plaza y permitiéndole visualizar la ubicación donde dejó su vehículo para hacer más simple la tarea de encontrarlo.

Los principales usuarios del sistema, serán los conductores de vehículos particulares. Quienes al momento de querer estacionar realizarán una consulta a través de su dispositivo móvil, la cual les permitirá saber dónde tienen estacionamiento libre cerca de su destino final, con las características de su agrado y como llegar hasta la misma.

Con la información recopilada, se podrá disponer de informes con datos como ubicación más requerida para estacionar, puntos críticos de la ciudad en materia de estacionamiento, horarios críticos. Estos informes le serán de utilidad a entidades como inversionistas de proyectos de estacionamientos y/o áreas del gobierno municipal como Tránsito.

## Impacto Ambiental

El **impacto ambiental** es el efecto que produce la actividad humana sobre el medio ambiente. Técnicamente, es la alteración de la línea de base ambiental.

Las acciones de las personas sobre el medio ambiente siempre provocarán efectos colaterales sobre éste. La preocupación por los impactos ambientales abarca varios tipos de acciones, como la contaminación, el consumo de energía, la contaminación acústica, la emisión de gases nocivos, o la pérdida de superficie de hábitats naturales, entre otros.

GeoParking es un proyecto que tiene un impacto positivo para el ambiente en los siguientes aspectos:

- **Emisión de Dióxido de Carbono:** la emisión de este gas ha crecido enormemente en los últimos tiempos debido al mayor consumo de combustibles por el crecimiento del número de vehículos que circulan sobre la superficie. El exceso de esta emisión de CO<sub>2</sub> es la que satura los niveles aceptables por la atmosfera y que por tal motivo se convierta en un factor de contaminación ambiental.

Con la utilización del sistema GeoParking, el recorrido de los vehículos en busca de plazas para estacionamientos será mucho menor a la que realizan actualmente los conductores. El saber que hayas tienen lugares disponibles para estacionar, y que además el conductor tiene a disposición la ruta más corta para llegar a él; es en su conjunto, un combo que permite que el menor recorrido de los vehículos signifique una menor emisión de dióxido de carbono en el ambiente.

Ya que una gran parte del tránsito que circula en los centros urbanos, es porque la gente se dirige hacia esos puntos a trabajar y por lo cual llega en sus vehículos particulares en

busca de estacionamiento, es lo que hace que la disminución de esa emisión contaminante sea de un gran beneficio para el ambiente de las ciudades.

- **Utilización de Energías:** En la actualidad, el incremento de la demanda y consumo de energía y las dificultades que existen para satisfacer esta demanda con las fuentes de energía disponibles, están prefigurando un escenario de crisis energética global. GeoParking produce que un menor traslado de los vehículos en busca de estacionamiento, lo cual significa un menor consumo de combustibles ya sea automóviles a nafta, gasoil o con equipos a gas.  
Otro punto es la energía que consumen los dispositivos móviles, pero si tenemos en cuenta que son dispositivos que todo el día están encendidos, el impacto de la utilización del sistema no provocara un mayor consumo de energía eléctrica, sino una reutilización eficiente de esa energía para producir un menor consumo de otras energías (combustibles).
- **Contaminación Acústica:** La contaminación acústica es considerada por la mayoría de la población de las grandes ciudades como un factor medioambiental muy importante, que incide de forma principal en su calidad de vida.  
La contaminación ambiental urbana o ruido ambiental es una consecuencia directa no deseada de las propias actividades que se desarrollan en las grandes ciudades. La constante circulación de vehículos en los centros urbanos produce altos niveles de ruido, los cuales producen un nivel importante de contaminación acústica al ambiente. Es por ello que si los automóviles que circulan en busca de estacionamiento, lo hacen por un tiempo mucho menor al habitual, se lograra disminuir la cantidad de autos en circulación y por lo tanto menor ruido en el ambiente.  
Esa es una de las maneras positivas en que influye GeoParking en cuanto al tiempo que tiene que circular un vehículo hasta encontrar un lugar donde estacionar.

## Impacto Social

La utilización de las nuevas tecnologías ha influido en el modo en que las personas viven el día a día. Lo que se trata en este apartado es demostrar cómo GeoParking trata de solucionar problemas e impactar de manera positiva en algunos hábitos que vive la gente en la actualidad, con referencia a la hora de manejarse en vehículo por los centros urbanos.

- **Congestión Vehicular:** en los centros urbanos en donde el tránsito vehicular es de envergadura, se producen congestiones en las calles de forma constante, y más si existen conductores que circulan a velocidades menores a la de los demás, solo por estar en busca de un lugar donde estacionar.  
Es por ello que GeoParking intenta impedir que esto suceda. El conductor podrá tener en claro su objetivo, es decir la playa en donde de seguro tendrá su lugar de estacionamiento y además la ruta por donde debe dirigirse hacia la misma.  
Todo esto propicia una circulación más fluida de los vehículos en busca de lugar donde estacionar.

- **Estrés de Conductores:** buscar un lugar donde estacionar, en una ciudad altamente invadida por automóviles circulando por todas las arterias de la misma, produce que el estrés del conductor aumente a medida que el tiempo pasa, y sigue conduciendo sin poder lograr el objetivo.

Además esto conlleva a que en momentos este conductor pueda producir lo que anteriormente hacíamos referencia (una congestión) y que esto le traiga inconvenientes con los demás conductores a los que le está obstruyendo o dificultando el paso.

Todo esto es posible de evitar con la utilización de GeoParking para encontrar un lugar seguro donde estacionar y la ruta precisa para acceder a ella.

**Disminución de costos de los estados municipales:** la municipalidad de Córdoba a través de la Comisión De Elaboración Del Plan Integral Y Estratégico De Movilidad De Córdoba en el Mayo de 2012, elaboro un pre-diagnóstico de movilidad donde, entre otras cosas dice,

*“A fines del año 2011 la normativa se extiende a los distritos 1, 2,3 y 4 (B°General Paz, Cofico, Alta Córdoba, etc.) para departamentos cuya superficie sea más de 100 m2 Hasta 150 m2, un espacio de estacionamiento cada unidad, y para departamentos cuya superficie sea superior a 150 m2 dos espacios de estacionamiento cada unidad.*

*El actual esquema de circulación, la ubicación de los centros atractores y un débil Control hace que la situación tenga un costo cada vez más oneroso, no sólo en términos de Tarifa a pagar por el vehículo particular, sino y muy especialmente en términos sociales.*

*Esto es así debido a que esta demanda de espacios que no encuentra suficiente Capacidad en la oferta, hace que quien se desplaza en vehículos individuales, permanezcan Circulando más allá de lo necesario, con una sobreocupación de las vías, dando como Resultado mayor costo en demoras para el conjunto de sus usuarios, por lo que solicitan Soluciones costosas para el Estado Municipal, ya que implican obras en soterramiento u Ocupación de más espacio público, para la demanda de mayor movilidad en vehículo Individual.”* Se puede notar el beneficio que el sistema traería a los municipios.

## Conclusión

Las tecnologías constituyen el núcleo central de una transformación que experimenta la sociedad y el ambiente que la rodea. De aquí lo importante es el estudio y dominio que tal transformación impone al ser humano, ya que tiende a modificar sus hábitos y patrones de vida, y su relación con el medio ambiente.

A través de este informe, hemos podido observar algunos de los beneficios que la implementación del sistema GeoParking traería a la comunidad, siendo estos de variadas categorías, ya que traería beneficios al medio ambiente; a los conductores de vehículos particulares, a los usuarios de las calles de la ciudad ya sean usuarios del transporte público, peatones, usuarios de bicicletas, usuarios de motocicletas, y demás usuarios de las vías públicas; y a los distintos estados municipales donde se implemente el sistema.

## Referencias

1er informe Pre-diagnóstico de Movilidad - Mayo de 2012 - Comisión De Elaboración Del Plan Integral De Movilidad De La Ciudad De Córdoba -

<http://www2.cordoba.gov.ar/portal/wp-content/uploads/downloads/2013/02/Prediagnostico.pdf>

# Informe de auditoria

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	
<b>Referencia:</b>	GeoP_Proyecto_Auditoria.docx
<b>Autores:</b>	Leonel Romero
<b>Fecha:</b>	23/08/2014

### Histórico de Versiones.

<b>Versión</b>	<b>Fecha</b>	<b>Estado</b>	<b>Responsable</b>	<b>Cambios</b>
1.0_DraftA	29/05/2014	Pendiente de Revisión	Leonel Romero [autor]	
1.0	04/09/2014	Aprobado	Lucas Toneatto	

## Introducción

El presente documento detalla la forma de trabajo y los resultados obtenidos en la auditoría de gestión de la configuración realizada el día jueves veintiuno de Agosto (21/08/2014) por el equipo de GeoParking.

La auditoría fue realizada por motivación propia del equipo, el objetivo principal fue establecer si el proyecto sigue los lineamientos plasmados en el documento de Gestión de la Configuración, el cual detalla las características de los distintos documentos almacenados en el repositorio, para realizar los cambios necesarios con el fin de realizar el primer release del sistema. Otro objetivo fue el de limpiar el repositorio, es decir, no almacenar archivos, información, software, etc. que el equipo no considera útil.

## Forma de Trabajo

La auditoría consto de 3 etapas realizadas en 3 días diferentes.

1. **Preparación:** Se revisaron todas las carpetas del repositorio, y se asignaron encargados a las distintas secciones del repositorio, se decidió realizarlo de esta manera para lograr realizar la auditoria de una manera más ágil.
2. **Auditoria:** Cada encargado audito su sección del repositorio detallando en un documento los problemas o cuestiones a tener en cuenta durante la discusión.
3. **Discusión:** Se reunió todo el equipo y se hizo una puesta en común de las distintas auditorías realizadas, luego tuvo lugar un espacio de discusión sobre cada punto a tratar, y se llegó a un acuerdo de las medidas a tomar.

## Resultados

### Auditoria de documentación

#### Capacitación

- De **GitHub** hay un archivo HTML, debería ser un archivo de documentación al igual que los demás.
  - *OBSERVACION DEL EQUIPO:* se empezara un nuevo documento el cual contendrá la información necesaria sobre el tema; y le mismo tendrá el formato correspondiente a los documentos de capacitación.
  - *ENCARGADO:* Ezequiel Bär Coch.
- Falta Documentación de **Ajax**.
  - *OBSERVACION DEL EQUIPO:* *empezar documentación*
  - *ENCARGADO:* Lucas Toneatto
- Sigue estando el material de **Android**, analizar si vale la pena que forme parte del proyecto si ya no es una de las alternativas a utilizar. No es conveniente que haya información por si acaso o por que puede servir.



- *OBSERVACION DEL EQUIPO*: se empezara un documento el cual contendrá la información necesaria sobre el tema; y le mismo tendrá el formato correspondiente a los documentos de capacitación.
  - *ENCARGADO*: Lucas Toneatto.
- De **Api de Google Maps** hay dos documentos que podrían ser uno. Además ver de qué manera almacenar los ejemplos de código.
  - *OBSERVACION DEL EQUIPO*: se unificaran los documentos referentes al tema.
  - *ENCARGADO*: Ignacio Frigerio.
- De **Entity Framework**: falta de actualizar temas utilizados en el proyecto. Caso carga diferida o no diferida y programada a mano. Además ver de qué manera almacenar los ejemplos de código.
  - *OBSERVACION DEL EQUIPO*: se completara la documentación correspondiente.
  - *ENCARGADO*: Leo Romero
- De la utilización de **Mensajes** en el proyecto, ver si está actualizada con referente a lo programado. Debería haber ejemplos.
  - *OBSERVACION DEL EQUIPO*: se completara la documentación con un ejemplo de utilización del código.
  - *ENCARGADO*: Leo Romero
- De **IceScrum** ver si sirve ese resumen que hay sobre producto. El PDF está en inglés. Puede haber documentación en varios idiomas o es preferible un mismo idioma. Sería mejor pasarlo a otro archivo, traducirlo y que tenga el formato de los demás archivos.
  - *OBSERVACION DEL EQUIPO*: se eliminara el archivo Word y se realizara un archivo de documentación específico y de referencia
  - *ENCARGADO*: Lucas Toneatto
- De **Phonegap** decidir si es la tecnología a utilizar o no, para ver si es necesario esa información de capacitación.
  - *OBSERVACION DEL EQUIPO*: se eliminará la documentación.
  - *ENCARGADO*: Lucas Toneatto
- De **Testing** no hay nada, solo un archivo de intento de un commit.
  - *OBSERVACION DEL EQUIPO*: eliminar txt, y completar con documentación
  - *ENCARGADO*: Leo Romero
- De **MVC** se debería eliminar esta carpeta y pasar lo de **LINQ** a otra carpeta específica para ello.
  - *OBSERVACION DEL EQUIPO*: se eliminara dicha carpeta
  - *ENCARGADO*: Lucas Toneatto

## Currículos

- ¿Cuál es la idea de esta carpeta?
  - *OBSERVACION DEL EQUIPO*: se eliminara dicha carpeta
  - *ENCARGADO*: Lucas Toneatto

## Manuales

- Hay un solo documento de **manual**, se debe actualizar y definir la forma de actualización de mismo para no tener problemas de sincronización, por si dos personas tratan de modificar el manual para hacer dos partes distintas.
  - *OBSERVACION DEL EQUIPO: se separa en manuales por roles*
  - *ENCARGADO: Marcos Barreras.*

## Minutas reuniones

- Definir si se van a seguir utilizando las **minutas de reuniones internas** o se dejan de usar y se descarta la carpeta.
  - *OBSERVACION DEL EQUIPO: se eliminar la carpeta y las próximas minutas será comunicadas y difundidas por Gmail.*

## Planes

- Solo está el de Testing el cual está incompleto. Deberían estar también los de Gestión de Riesgos, Gestión de la Configuración, etc.
  - *OBSERVACION DEL EQUIPO: Se completaran y agregaran los documentos correspondientes.*
  - *ENCARGADO: Leo Romero*

## Auditoria de informes de reunión

### Reuniones de clase

- Están faltando 5 documentos (Mayo: 13 – 20 – 27 Junio: 3 Agosto: 5 )  
¿Porque faltan? ¿Están en otro lado? ¿Todos esos días tuvimos clases?
  - *OBSERVACION DEL EQUIPO: 13/5 parcial, 20/5 presentación, 27/5 presentación, 3/6 desaparecido en acción, 5/8 Buscar porque tiene que estar.*

### Retrospectives

- Tenemos 2 modelos de Sprints diferentes, ¿dejamos los dos?
  - *OBSERVACION DEL EQUIPO: queda todo como esta*

## Auditoria de release

- Fechas de Retrospective no concuerdan con fechas de Icescrum.
  - *OBSERVACION DEL EQUIPO: sincronizar con icescrum.*
  - *Encargado: Marcos*
- No está la misma información en las carpetas para la profesora y en el repositorio.
  - *OBSERVACION DEL EQUIPO: sincronizar y completar*
  - *Encargado: Marcos*
- Carpeta de entregables, información que se muestra.

- *OBSERVACION DEL EQUIPO*: Dejar como esta

### **Icescrum**

- Duraciones inconsistentes de los Sprints en el Icescrum.
  - *OBSERVACION DEL EQUIPO*: sincronizar con retrospective.
  - *Encargado*: Marcos
- Información errónea en la generación de BurndownCharts.
  - *OBSERVACION DEL EQUIPO*: Se dejan como están, se explica que paso
- Resúmenes de Sprint incompletos
  - *OBSERVACION DEL EQUIPO*: Antes de cerrar el sprint, generamos los resúmenes
- El Sprint 6 ya termino pero nunca se cerró, por ende el sprint 7 que tendría que terminar hoy (21/08) todavía nunca empezó.
  - *OBSERVACION DEL EQUIPO*: queda como esta.

## **Auditoria de Software, Matriz de Trazabilidad y Product Backlog**

### **Software**

- De **AnkhSvn** ¿Para qué está?
  - *OBSERVACION DEL EQUIPO*: Eliminar
- De **Tortoise** ¿Para qué está?
  - *OBSERVACION DEL EQUIPO*: Eliminar
- De **StarUML** ¿Lo usamos?
  - *OBSERVACION DEL EQUIPO*: Eliminar
- De **VisualHG** ¿Para qué?
  - *OBSERVACION DEL EQUIPO*: Eliminar
- De **WBSChart** ¿Lo vamos a seguir usando?
  - *OBSERVACION DEL EQUIPO*: No lo utilizaremos más pero lo mantendremos para mantenimiento de wbs.
- Hace falta tener el **GitHubSetup**? Vale la pena?
  - *OBSERVACION DEL EQUIPO*: Lo mantendremos.

**Responsable: Eze**

### **Matriz de trazabilidad**

- Debería actualizarse los links de los tildes
  - *OBSERVACION DEL EQUIPO*: Si
- Se requiere cargar todas las demás historias de usuario y su traza
  - *OBSERVACION DEL EQUIPO*: si
- Vamos a asignar un responsable para esto?
  - *OBSERVACION DEL EQUIPO*: Si, se definirá luego.

**Responsable:** El scrum master

### Product backlog

- ¿Cuál es la idea de tener este Excel?
  - *OBSERVACION DEL EQUIPO: Permitía calcular las hs y no teníamos una herramienta.*
- ¿Vamos a seguir usando IceScrum y sincronizarlo con este Excel?
  - *OBSERVACION DEL EQUIPO: Vamos a usar IceScrum, a este Excel no.*
- ¿El product backlog que pidió la profe, habría que hacer otro nuevo?
  - *OBSERVACION DEL EQUIPO: Si*
- ¿Vamos a agregarle la tarea al Scrum master de turno que lleve al día este product backlog? ¿Y el IceScrum?
  - *OBSERVACION DEL EQUIPO: Que mantenga la herramienta. Que se encargue a pedirle a los miembros del equipos, gráficos, resumen del sprint que estará reflejado en el “PROCEDIMIENTO DEL SCRUM MASTER”*

**Responsable:** Scrum Master

- ¿Nuestro problema es la herramienta o **somos nosotros**?
  - *OBSERVACION DEL EQUIPO: Si por eso estaremos más precisos y atentos*
- ¿Quién va a hacer el Excel para la profe?
  - *OBSERVACION DEL EQUIPO: El Scrum Master*

### Auditoria de Templates

- El template de casos de prueba esta desactualizado, ¿Qué hacemos?
  - *OBSERVACION DEL EQUIPO: Modificarlo para adecuarlo al nuevo estilo de CP*
  - *Responsable: Ignacio*
- Está faltando un template de ejecución de casos de prueba
  - *OBSERVACION DEL EQUIPO: Crearlo*
  - *Responsable: Ignacio*
- Falta darle estilos al template de estimaciones.
  - *OBSERVACION DEL EQUIPO: Modificar*
  - *Responsable: Ignacio*

## Conclusión

La realización de la auditoria demostró al equipo, que se viene trabajando de una manera correcta y en la mayoría de los casos consistente con el documento de Gestión de la Configuración, como resultado de se ha decidido eliminar ciertos documentos, como así también reubicar otros.

Todos los miembros del equipo quedaron con tareas para realizar con el objetivo de cumplir con las decisiones de esta auditoría, y así poder poner fecha al primer release de GeoParking.

Es entonces que decidimos poner como plazo para realizar las acciones correspondientes el día jueves 28/08.

# Product Backlog

## Control de la documentación

### Control de la Configuración.

<b>Título:</b>	Product Backlog
<b>Referencia:</b>	GeoP_Proyecto_ProductBacklog.docx
<b>Autores:</b>	Lucas Toneatto
<b>Fecha:</b>	29/05/2014

### Histórico de Versiones.

<b>Versión</b>	<b>Fecha</b>	<b>Estado</b>	<b>Responsable</b>	<b>Cambios</b>
1.0_DraftA	29/05/2014	Pendiente de Revisión	Lucas Toneatto [autor]	
1.0	31/05/2014	Aprobado	Leonel Romero [Revisor]	
2.0_DraftA	26/01/2015	Pendiente de revisión	Marcos Barrera	Cambios importantes en todo el documento, organización y estilos.
2.0	28/01/2015	Aprobado	Ezequiel Bär Coch	
2.1_DrafA	22/02/2015	Pendiente de revisión	Marcos Barrera	Se actualizo el estado de las historias
2.1	26/02/2015	Aprobado	Ignacio Frigerio	

## Introducción

En este documento, queremos presentar las historias de usuario al día de la fecha identificadas por el equipo en el rol de Product Owner colectivo que definimos en la presentación del proyecto.

Atte. El Equipo de GeoParking

## Product Backlog

ID	HISTORIA	DESCRIPCION	ESFUERZO	PRIORIDAD	DEPENDENCIAS	ESTADO
1	Consultar playas de estacionamiento en el móvil (MAPA)	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> consultar las playas de estacionamiento ubicadas en la ciudad donde estoy, <b>para</b> elegir la más conveniente.	5.00	Alta	3	REALIZADA
2	Consultar playas de estacionamiento en la web	<b>Como</b> usuario web <b>Yo puedo</b> conocer las playas de estacionamiento cercanas a un lugar de mi interés <b>para</b> seleccionar aquella que sea más conveniente entre ellas	3.00	Alta	3	REALIZADA
3	Administrar playas de estacionamiento	<b>Como</b> usuario web <b>Yo puedo</b> registrar una nueva playa de estacionamiento <b>para</b> ser consultada por otro usuario	8.00	Alta	-	REALIZADA
4	Almacenar ubicación de vehículo	<b>Como</b> usuario de la aplicación móvil <b>Yo puedo</b> guardar la posición de mi vehículo <b>para</b> ser consultada posteriormente	1.00	Alta	-	REALIZADA
5	Consultar ruta a playa de estacionamiento	<b>Como</b> usuario de la aplicación móvil <b>Yo puedo</b> obtener la ruta a una playa de estacionamiento seleccionada <b>para</b> llegar a la misma a estacionar mi vehículo	3.00	Alta	3,4	REALIZADA

6	Consultar playas de estacionamiento cercanas a un punto de interés	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> consultar las playas cercanas a un punto de interés <b>para</b> conocer su ubicación y disponibilidad	5.00	Alta	3,26	REALIZADA
7	Consultar disponibilidad de playa de estacionamiento	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> consultar la disponibilidad de una playa de estacionamiento seleccionada <b>para</b> saber si puedo estacionar mi vehículo	3.00	Alta	3,8	REALIZADA
8	Actualizar la disponibilidad de una playa de estacionamiento	<b>Como</b> empleado de la playa <b>yo puedo</b> actualizar la disponibilidad de una playa de estacionamiento <b>para</b> que sea consultada por los usuarios de la aplicación móvil	5.00	Alta	3	REALIZADA
9	Consultar ruta a posición almacenada de vehículo	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> consultar la ruta a la posición de mi vehículo almacenada en mi móvil <b>para</b> volver al mismo	5.00	Alta	4	REALIZADA
10	Almacenar playas de estacionamiento consultadas	<b>Como</b> usuario de la aplicación móvil <b>yo quiero</b> que se almacenen las playas consultadas	3.00	Alta	3	REALIZADA



11	Administrar configuración de dispositivo	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> guardar mis configuraciones <b>para</b> que la aplicación sea utilizada en otro momento de acuerdo a mis preferencias	5.00	Alta	-	REALIZADA
12	Utilizar comando por voz para localización	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> utilizar la opción de comando por voz <b>para</b> buscar un punto de interés en el mapa.	34.00	Baja	-	PENDIENTE
13	Consultar disponibilidad de playas de estacionamiento en tiempo real	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> consultar la disponibilidad de una playa de estacionamiento en tiempo real <b>para</b> que la información obtenida sea consistente y actual.	5.00	Alta	-	REALIZADA
14	Crear reportes estadísticos (EPICA)	<b>Como</b> súper usuario <b>yo puedo</b> obtener reportes e informes pertinentes para el procesamiento del mismo] <b>para</b> una posterior toma de decisiones.	34.00	Media	1,2,3,5,6,7,8,10,11,13,26	REALIZADA
15	Gestionar permisos de usuarios	<b>Como</b> súper administrador de la página web <b>yo puedo</b> gestionar perfiles de usuarios y los permisos de los mismos <b>para</b> asignarlos a los usuarios existentes.	2.00	Alta	-	REALIZADA

16	Registrar usuario	<b>Como</b> usuario de la aplicación web <b>yo quiero</b> registrarme <b>para</b> acceder al sistema.	1.00	Alta	-	REALIZADA
17	Panel de Usuario	<b>Como</b> usuario del sistema <b>yo puedo</b> acceder a mi panel de usuario <b>para</b> poder hacer modificaciones de los datos personales.	3.00	Alta	-	REALIZADA
18	Consultar playas de estacionamiento en el móvil (Listado)	<b>Como</b> usuario de la aplicación móvil <b>yo puedo</b> consultar las playas de estacionamiento cercanas a mi posición.	5.00	Alta		REALIZADA
19	Configurar Servidor	Configuración de servidor que albergara la aplicación web y el web Service.	3.00	Alta	-	REALIZADA
20	Consultar historial de playas	<b>Como</b> usuario de la aplicación <b>yo puedo</b> ver las ultimas playas consultadas.	2.00	Alta	-	REALIZADA
21	Gestionar permisos de roles	<b>Como</b> usuario de la aplicación web <b>yo puedo</b> asignar permisos a roles existentes.	2.00	Alta	-	REALIZADA
22	Creación de roles	<b>Como</b> usuario de la aplicación web <b>yo quiero</b> crear roles de usuario <b>para</b> administrar la seguridad de acceso.	2.00	Alta	-	REALIZADA

23	Administración de zonas	Como administrador <b>yo quiero</b> registrar, eliminar o modificar zonas de interés.	3.00	Alta	-	REALIZADA
24	Visualizar estadísticas en tiempo real	Como administrador <b>yo quiero</b> ver estadísticas en tiempo real de la disponibilidad y consultas.	3.00	Alta	-	REALIZADA
25	Visualizar estadísticas históricas	Como administrador, <b>yo quiero</b> ver estadísticas históricas de disponibilidad y consultas.	3.00	Alta	-	REALIZADA
26	Cambiar tipos de gráficos	Como administrador, <b>yo quiero</b> cambiar los gráficos en los que se muestran las estadísticas.	3.00	Alta	-	REALIZADA
27	Filtrar estadísticas	Como administrador, <b>yo quiero</b> filtrar las estadísticas que están siendo visualizadas.	3.00	Alta	-	REALIZADA
28	Comparar estadísticas	Como administrador, <b>yo quiero</b> agregar una nueva representación de estadísticas <b>para</b> poder comparar.	3.00	Alta	-	REALIZADA
29	Administrar entidades básicas	Como administrador, <b>yo quiero</b> administrar las entidades básicas (tipo de playa, tipo de vehículo, días de atención y tiempos).	3.00	Alta	-	PENDIENTE

## Detalle de historias de usuario

<b>ID: 1</b>	<b>NOMBRE: Consultar playas de estacionamiento en el móvil (MAPA)</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil, yo puedo consultar las playas de estacionamiento ubicadas en la ciudad donde estoy, para elegir la más conveniente.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Mostrar Playas en Ciudad</u></b> Dada la ciudad en que me ubico, Cuando existen playas de estacionamiento, Entonces se muestran las mismas en un mapa.</p> <p style="text-align: center;"><b><u>Mostrar mensaje por falta de playas</u></b> Dada la ciudad en que me ubico, Cuando no existen playas de estacionamiento, Entonces se muestra un mensaje informando la situación.</p>	

<b>ID: 2</b>	<b>NOMBRE: Consultar playas de estacionamiento en la web</b>
<b>DESCRIPCION:</b> Como usuario web, yo puedo conocer las playas de estacionamiento cercanas a un lugar de mi interés para obtener información de las mismas.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Búsqueda de playa de estacionamiento por punto de referencia</u></b> Dado que existen playas de estacionamiento que cumplen con los filtros seleccionados Cuando busco un punto de referencia Entonces se muestran las playas de estacionamiento cercanas al punto de referencia dentro de un radio determinado.</p> <p style="text-align: center;"><b><u>Búsqueda de playa de estacionamiento por ciudad</u></b> Dado que existen playas de estacionamiento en la ciudad a buscar Cuando selecciono una ciudad Entonces se muestra las playas de estacionamiento de esa ciudad en mapa y sus detalles en una grilla.</p> <p style="text-align: center;"><b><u>Búsqueda por filtro</u></b> Dado que existan playas de estacionamiento cargadas que cumplen con los filtros correspondientes (Dirección, Tipo de Playa, Tipo de Vehículo, Precio, Día y Horario de Atención.) Cuando filtro la búsqueda Entonces se muestran las playas de estacionamiento en el mapa que cumplen con el criterio de filtrado.</p>	

<b>ID: 3</b>	<b>NOMBRE: Administrar Playas de Estacionamiento</b>
<b>DESCRIPCION:</b> Como usuario web yo puedo registrar, editar, ver y eliminar una playa de estacionamiento.	
<b>PRUEBAS DE ACEPTACION:</b> <p style="text-align: center;"><b><u>Alta de playa de estacionamiento (Éxito)</u></b></p> <p style="text-align: center;">Dado una playa de estacionamiento que no existe en la base de datos  Cuando ingreso todos sus datos y se registra  Entonces se muestra un mensaje de éxito.</p> <p style="text-align: center;"><b><u>Alta de playa de estacionamiento (Fracaso)</u></b></p> <p style="text-align: center;">Dado que la playa de estacionamiento no existe en la base de datos  Cuando no ingreso todos aquellos datos que son obligatorios e intento registrarla  Entonces se muestran mensajes de error indicando los datos necesarios e incompletos.</p> <p style="text-align: center;"><b><u>Baja de playa de estacionamiento</u></b></p> <p style="text-align: center;">Dado que existe una playa de estacionamiento  Cuando borro la playa  Entonces entonces se muestra un mensaje confirmando la acción.</p> <p style="text-align: center;"><b><u>Consulta de playa de estacionamiento por nombre</u></b></p> <p style="text-align: center;">Dado que existe la playa de estacionamiento y conozco el nombre de la misma.  Cuando quiero buscarla ingresando dicho nombre  Entonces se muestra la información detallada de la misma.</p> <p style="text-align: center;"><b><u>Modificación de datos de playa</u></b></p> <p style="text-align: center;">Dado que existe la playa de estacionamiento a modificar y cuento con los datos a reemplazar  Cuando ingreso los datos y presionar el botón Guardar cambios  Entonces se muestra un mensaje de éxito</p>	

<b>ID: 4</b>	<b>NOMBRE: Almacenar ubicación vehículo</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil, yo puedo guardar la posición de mi vehículo para ser consultada posteriormente.	
<b>PRUEBAS DE ACEPTACION:</b> <p style="text-align: center;"><b><u>Guardar Posición Vehículo</u></b></p> <p style="text-align: center;">Dado que deseo guardar la posición actual de mi vehículo  Cuando guardo dicha posición  Entonces se muestra un mensaje de éxito.</p>	

<b>ID: 5</b>	<b>NOMBRE: Consultar ruta a playa de estacionamiento</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil, yo puedo obtener la ruta a una playa de estacionamiento seleccionada para llegar a la misma.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Mostrar Ruta a Playa</u></b>  Dada la posición actual del vehículo,  Cuando selecciono la playa destino,  Entonces se traza la ruta y se actualice la misma cada 3 segundos.</p> <p style="text-align: center;"><b><u>Dejar de Mostrar Ruta a Playa</u></b>  Dada la posición actual, y la de la playa destino,  Cuando la distancia entre la posición actual y la de playa es menor a 20 metros  Entonces la ruta desaparece del mapa.</p>	

<b>ID: 6</b>	<b>NOMBRE: Consultar playas de estacionamiento cercanas a un punto de interés</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil, yo puedo consultar las playas cercanas a un punto de interés para conocer su ubicación.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Consultar Playas cercanas a punto de interés</u></b>  Dada un punto de interés  Cuando lo ingreso  Entonces se muestran las playas cercanas al punto de interés.</p>	

<b>ID: 7</b>	<b>NOMBRE: Consultar disponibilidad de playa de estacionamiento</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil, yo puedo consultar la disponibilidad de una playa de estacionamiento seleccionada para saber si puedo estacionar mi vehículo	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Mostrar disponibilidad de Playa(Éxito)</u></b>  Dado que existen playas de estacionamiento cercanas a mi ubicación  Cuando las visualizo  Entonces me muestra la disponibilidad actual de cada una de las playas.</p>	

<b>ID: 8</b>	<b>NOMBRE: Actualizar disponibilidad de una playa de estacionamiento</b>
<b>DESCRIPCION:</b> Como empleado de la playa, yo puedo actualizar la disponibilidad de una playa de estacionamiento para que sea consultada por los usuarios de la aplicación móvil.	
<b>PRUEBAS DE ACEPTACION:</b> <u><i>Actualizar disponibilidad Playa por Ingreso/Egreso de vehículos</i></u> Dado el ingreso o egreso de un vehículo a la playa Cuando registro dicho movimiento Entonces se actualiza la disponibilidad de la playa de estacionamiento.	

<b>ID: 9</b>	<b>NOMBRE: Consultar ruta a posición almacenada de vehículo</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil, yo puedo consultar la ruta a la posición de mi vehículo almacenada para volver al mismo.	
<b>PRUEBAS DE ACEPTACION:</b> <u><i>Consultar ruta a vehículo</i></u> Dada mi posición actual y teniendo guardada la posición donde estacione mi vehículo Cuando quiero regresar a la misma Entonces me traza la ruta de regreso en el mapa.	

<b>ID: 11</b>	<b>NOMBRE: Administrar configuración de dispositivo</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil yo puedo guardar mis configuraciones para que la aplicación sea utilizada de acuerdo a mis preferencias.	
<b>PRUEBAS DE ACEPTACION:</b> <u><i>Guardar configuración de dispositivo</i></u> Dadas mis preferencias Cuando las ingreso en la aplicación Entonces se almacenan y son utilizadas por la misma	

<b>ID: 12</b>	<b>NOMBRE: Utilizar comando por voz para localización</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil yo puedo utilizar la opción de comando por voz para buscar un punto de interés en el mapa.	
<b>PRUEBAS DE ACEPTACION:</b> <u><i>Buscar playas cercanas a punto de interés por voz</i></u> Dado que ingreso un punto de interés por comando de voz Cuando se reconoce el mismo Entonces se muestra el punto de interés y las playas cercanas a él.	

<b>ID: 13</b>	<b>NOMBRE: Consultar disponibilidad de playas de estacionamiento en tiempo real.</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil, yo puedo consultar la disponibilidad de una playa de estacionamiento en tiempo real, para que la información obtenida sea consistente y actual.	
<b>PRUEBAS DE ACEPTACION:</b> <p style="text-align: center;"><b><u>Consultar disponibilidad actual</u></b>  Dada una playa seleccionada  Se debe realizar una actualización cada 2 segundos de la disponibilidad  Y mostrarme la misma en la interfaz.</p>	

<b>ID: 14</b>	<b>NOMBRE: Crear reportes estadísticos (EPICA)</b>
<b>DESCRIPCION:</b> Como superusuario, yo puedo obtener reportes e informes pertinentes para el procesamiento del mismo para una posterior toma de decisiones	
<b>PRUEBAS DE ACEPTACION:</b>	

<b>ID: 15</b>	<b>NOMBRE: Gestionar roles de usuarios</b>
<b>DESCRIPCION:</b> Como súper administrador de la página web yo puedo gestionar perfiles de usuarios y los permisos de los mismos para asignarlos a los usuarios existentes.	
<b>PRUEBAS DE ACEPTACION:</b> <p style="text-align: center;"><b><u>Asignación de rol a usuario</u></b>  Dado que existe un usuario y un rol  Cuando quiero asignar un rol al usuario  Entonces se muestra un mensaje de éxito.</p>	



<b>ID: 16</b>	<b>NOMBRE: Registrar Usuario</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación web yo quiero registrarme para acceder al sistema.	
<b>PRUEBAS DE ACEPTACION:</b>	
<p style="text-align: center;"><b><u>Creación de usuario (Éxito)</u></b></p> <p style="text-align: center;"><i>Dado que el usuario no existe</i></p> <p style="text-align: center;"><i>Cuando completo los campos requeridos: nombre de usuario, contraseña, email (siendo todos válidos) y se registra.</i></p> <p style="text-align: center;"><i>Entonces se muestra un mensaje de éxito y se envía un correo electrónico de confirmación.</i></p> <p style="text-align: center;"><b><u>Creación de usuario (Fracaso)</u></b></p> <p style="text-align: center;"><i>Dado que el usuario no existe</i></p> <p style="text-align: center;"><i>Cuando completo los campos requeridos: nombre de usuario, contraseña, email (incorrectos) y se registra.</i></p> <p style="text-align: center;"><i>Entonces se muestra un mensaje informando la situación.</i></p>	

<b>ID: 17</b>	<b>NOMBRE: Panel de usuario</b>
<b>DESCRIPCION:</b> Como usuario del sistema yo puedo acceder a mi panel de usuario para poder hacer modificaciones de los datos personales.	
<b>PRUEBAS DE ACEPTACION:</b>	
<p style="text-align: center;"><b><u>Modificar usuario</u></b></p> <p style="text-align: center;"><i>Dado que el usuario existe en el sistema, esta logueado y quiere modificar su información</i></p> <p style="text-align: center;"><i>Cuando ingresa al panel de usuario y modifica sus datos</i></p> <p style="text-align: center;"><i>Se actualizan sus datos y se muestra un mensaje de éxito.</i></p>	

<b>ID: 18</b>	<b>NOMBRE: Consultas de playas de estacionamiento en el Móvil (LISTADO)</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación móvil yo puedo consultar las playas de estacionamiento cercanas a mi posición.	
<b>PRUEBAS DE ACEPTACION:</b>	
<p style="text-align: center;"><b><u>Listar playas de estacionamiento cercanas (Éxito)</u></b></p> <p style="text-align: center;"><i>Dado que existen playas de estacionamiento cercanas a mi posición actual</i></p> <p style="text-align: center;"><i>Cuando quiero visualizar el listado</i></p> <p style="text-align: center;"><i>Entonces se muestra el mismo con las playas e información básica de cada una.</i></p>	

<b>ID: 19</b>	<b>NOMBRE: Configurar Servidor</b>
<b>DESCRIPCION:</b> Configuración de servidor que albergara la aplicación web y el web Service.	
<b>PRUEBAS DE ACEPTACION:</b>  Dado que la aplicación y el web service están deployados correctamente. Cuando quiero hacer consultas vía Ajax Entonces el web service debe responder con los datos solicitados.	

<b>ID: 20</b>	<b>NOMBRE: Consultar historial de playas</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación yo puedo ver las ultimas playas consultadas.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Consultar historial</u></b></p> Dado que existen playas almacenadas Cuando quiero ver el listado de las mismas Entonces se muestra las últimas playas consultadas recientemente.	

<b>ID: 21</b>	<b>NOMBRE: Gestionar permisos de roles</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación web yo puedo asignar permisos a roles existentes.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Asignación de permisos a roles</u></b></p> Dado que existen roles y permisos, Cuando se asigna uno o más permisos al rol Entonces se muestra un mensaje de asignación correcta.	

<b>ID: 22</b>	<b>NOMBRE: Creación de roles</b>
<b>DESCRIPCION:</b> Como usuario de la aplicación web yo quiero crear roles de usuario para administrar la seguridad de acceso.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Creación de roles</u></b>  <i>Dado que el rol no existe en la BD y quiero crear uno nuevo</i>  <i>Cuando presiono el botón "Guardar"</i>  <i>Entonces se crea un nuevo rol con el nombre que le pusimos.</i></p>	

<b>ID: 23</b>	<b>NOMBRE: Administración de zonas</b>
<b>DESCRIPCION:</b> Como administrador quiero registrar, eliminar o modificar zonas de interés.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Creación de zonas</u></b>  <i>Dado que la zona no existe,</i>  <i>Cuando la registro</i>  <i>Entonces se muestra un mensaje informando la situación.</i></p>	

<b>ID: 24</b>	<b>NOMBRE: Visualizar estadísticas en tiempo real</b>
<b>DESCRIPCION:</b> Como administrador quiero ver estadísticas en tiempo real de la disponibilidad y consultas.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Actualización de estadísticas</u></b>  <i>Dado que hay usuarios utilizando el sistema</i>  <i>Cuando pasa un tiempo determinado</i>  <i>Entonces se actualizan las estadísticas.</i></p>	

<b>ID: 25</b>	<b>NOMBRE: Visualizar estadísticas históricas</b>
<b>DESCRIPCION:</b> Como administrador quiero ver estadísticas históricas de disponibilidad y consultas.	
<b>PRUEBAS DE ACEPTACION:</b>  <p style="text-align: center;"><b><u>Ver estadísticas</u></b>  <i>Dado que hay datos cargados</i>  <i>Cuando quiero ver las estadísticas</i>  <i>Entonces se visualizan.</i></p>	

<b>ID: 26</b>	<b>NOMBRE: Cambiar tipos de gráficos</b>
<b>DESCRIPCION:</b> Como administrador, quiero cambiar los gráficos en los que se muestran las estadísticas.	
<b>PRUEBAS DE ACEPTACION:</b>	
<p style="text-align: center;"><b><u>Cambio de grafico</u></b>  <i>Dado</i> que se está visualizando una estadística  <i>Cuando</i> quiero cambiar su forma de visualización  <i>Entonces</i> se modifica la forma en la que se representa.</p>	

<b>ID: 27</b>	<b>NOMBRE: Filtrar estadísticas</b>
<b>DESCRIPCION:</b> Como administrador, quiero filtrar las estadísticas que están siendo visualizadas.	
<b>PRUEBAS DE ACEPTACION:</b>	
<p style="text-align: center;"><b><u>Filtrado de estadísticas</u></b>  <i>Dado</i> que se están visualizando estadísticas  <i>Cuando</i> aplico filtros  <i>Entonces</i> se representa la estadística para los filtros aplicados.</p>	

<b>ID: 28</b>	<b>NOMBRE: Comparar estadísticas</b>
<b>DESCRIPCION:</b> Como administrador, quiero agregar una nueva representación de estadísticas para poder comparar.	
<b>PRUEBAS DE ACEPTACION:</b>	
<p style="text-align: center;"><b><u>Agregar Comparación</u></b>  <i>Dado</i> que se están visualizando estadísticas  <i>Cuando</i> quiero agregar una comparación  <i>Entonces</i> se agrega una nueva representación de estadísticas.</p>	

<b>ID: 29</b>	<b>NOMBRE: Administrar entidades básicas</b>
<b>DESCRIPCION:</b> Como administrador, quiero administrar las entidades básicas (tipo de playa, tipo de vehículo, días de atención y tiempos).	
<b>PRUEBAS DE ACEPTACION:</b>	
<p style="text-align: center;"><b><u>Creación de entidades</u></b>  <i>Dado</i> que la entidad no existe,  <i>Cuando</i> la registro  <i>Entonces</i> se muestra un mensaje informando la situación.</p>	

## ANEXO A

#	RIESGO	AMBITO	CONOCIMIENTO	CONTEXTO	MOMENTO	PROBABILIDAD	IMPACTO	RIESGO
1	Mala estimación de tiempos	PROYECTO	PREDECIBLE	INTERNO	TODOS	ALTA	ALTO	ALTO
2	Falta de conocimiento técnico	TECNICO	PREDECIBLE	INTERNO	DESARROLLO	MEDIA	MEDIO	MEDIO
3	Problemas o fallas técnicas en el repositorio.	TECNICO	IMPREDECIBLE	EXTERNO	DESARROLLO	BAJA	ALTO	ALTO
4	No se cumple con los plazos de entrega previstos	PROYECTO	PREDECIBLE	INTERNO	TODOS	MEDIA	MEDIO	MEDIO
5	El sistema no cumple con las expectativas de los usuarios	NEGOCIO	PREDECIBLE	INTERNO	IMPLEMENTACION	BAJA	BAJA	BAJO
6	Sobreasignación de tareas al equipo	PROYECTO	PREDECIBLE	INTERNO	TODOS	BAJA	BAJA	BAJO
7	Falta de compromiso del equipo	PROYECTO	PREDECIBLE	INTERNO	TODOS	BAJA	ALTO	MEDIO
8	Subestimar el alcance del sistema	PROYECTO	PREDECIBLE	INTERNO	PLANEACION	MEDIA	ALTO	ALTO
9	Falta de liderazgo en el proyecto	PROYECTO	PREDECIBLE	INTERNO	TODOS	MEDIA	MEDIO	MEDIO
10	No cumplir todas las funcionalidades del producto	TECNICO	PREDECIBLE	INTERNO	IMPLEMENTACION	BAJA	ALTO	ALTO
11	Subestimar entregables de proyecto	PROYECTO	PREDECIBLE	INTERNO	DESARROLLO	BAJA	BAJA	BAJO
12	No se pudo implementar el sistema en Android	TECNICO	IMPREDECIBLE	INTERNO	DESARROLLO	BAJA	MEDIO	BAJO
13	No se puede Realizar comunicación con Software de Playa de estacionamiento	TECNICO	IMPREDECIBLE	EXTERNO	IMPLEMENTACION	ALTA	ALTO	ALTO