
Plan de Gestión de Proyecto

Proyecto:

Revisión 1.0






Ficha del documento

Fecha	Revisión	Autor	Verificado
25/04/2018	1.0	 Island	

Documento validado por las partes en fecha:

Por el cliente	Por la empresa suministradora
	
Aclaración:	Aclaración: Island



Contenido

FICHA DEL DOCUMENTO	2
CONTENIDO	3
1 INTRODUCCIÓN	5
1.1 Resumen del proyecto	5
1.1.1 Propósito, alcance y objetivos	5
1.1.2 Supuestos y restricciones	5
1.1.3 Entregables del proyecto	5
1.1.4 Calendario y resumen del presupuesto	5
2 DOCUMENTOS REFERENCIADOS	6
3 ORGANIZACIÓN DEL PROYECTO	7
3.1 Interfaces externas	7
3.2 Estructura Interna	7
3.3 Roles y responsabilidades	7
4 PLANES DE ADMINISTRACIÓN DE PROCESO	8
4.1 Plan inicial	8
4.1.1 Plan de personal	8
4.1.2 Plan de adquisición de recursos	8
4.1.3 Plan de entrenamiento del personal del Proyecto	8
4.2 Plan de trabajo	9
4.2.1 Principales actividades del proyecto	9
4.2.2 Asignación de esfuerzo	10
4.2.3 Asignación de presupuesto	10



4.3	Plan de administración de riesgos	10
5	PLANES DE PROCESOS TÉCNICOS	11
5.1	Modelo de proceso	11
5.2	Métodos, herramientas y técnicas	12
5.3	Plan de infraestructura	12
5.4	Plan de aceptación del producto	12
6	PLAN DE PROCESOS DE APOYO	12
6.1	Plan de documentación	12
7	PLANES ADICIONALES	13



1. Introducción

1.1. Resumen del Proyecto

1.1.1. Propósito, alcance y objetivos.

El propósito del software a entregar, Un Aventón!, es poder generar viajes compartidos entre diferentes usuarios y compartir el gasto de los mismos.

El alcance del producto está definido a todas las personas que estén interesadas en la aplicación, pero principalmente a los ideólogos de la misma.

1.1.2. Supuestos y restricciones

El proyecto tiene fecha de entrega final el Jueves 02 de Agosto del corriente año (02/08/2018).

Como restricciones se debe considerar el uso de una interfaz de usuario intuitiva, la utilización de los colores del logo de Un Aventón! para la página web, y las restricciones dadas por el lenguaje elegido para desarrollarla.

El lenguaje de programación elegido para desarrollar la aplicación es Ruby, con el framework Rails.

Se utilizarán gemas de Ruby para diversas funcionalidades, como el manejo de la base de datos, verificación de datos, encriptación, etc.

Los recursos físicos (ejemplo máquinas para desarrollar el proyecto) están dados por la tecnología que posee cada integrante del proyecto.

1.1.3. Entregables del proyecto

Elementos a Entregar	Fecha
Demo 1 (Sprint 1 con Scrum diario 1)	31/05/2018
Demo 2 (Sprint 2 con Scrum diario 2)	28/06/2018
Demo 3 (Sprint 3 con Scrum diario 3)	02/08/2018

1.1.4. Calendario y resumen del presupuesto

El presupuesto total para el desarrollo de la aplicación será detallado en el punto 4.2.3 .

El tiempo total de desarrollo del sistema consta de 21 semanas, habiendo empezado el Jueves 15/03/2018, con fecha final de entrega el Jueves 02/08/2018.

Tiempos estimados:



Fecha	Actividad
15/03/2018	Primer entrevista
22/03/2018	Segunda entrevista
29/03/2018	Entrega de SRS y Entrevistas
05/04/2018	Construcción de Pila de Producto y DER
12/04/2018	Construcción de Pila de Producto y DER
19/04/2018	Entrega Pila de Producto y DER
26/04/2018	Entrenamiento del Personal
03/05/2018	Entrenamiento del Personal
10/05/2018	Planificación Sprint 1 Scrum diario 1
17/05/2018	Scrum diario 1
24/05/2018	Scrum diario 1
31/05/2018	DEMO 1
07/06/2018	Planificación Sprint 2 Scrum diario 2
14/06/2018	Scrum diario 2
21/06/2018	Scrum diario 2
28/06/2018	DEMO 2
05/07/2018	Planificación Sprint 3 Scrum diario 3
12/07/2018	Scrum diario 3
19/07/2018	Scrum diario 3
26/07/2018	Scrum diario 3
02/08/2018	DEMO 3

2. Documentos referenciados

Referencia	Título	Fecha	Autor
6	SRS v1.1	27/03/18	Island
4	Entrevista 1 v1.1	27/03/18	Island
5	Entrevista 2 v1.1	27/03/18	Island
7	DER	16/04/18	Island
9	Anexo de tratamiento de riesgos	29/04/18	Island
0	Plantilla PGP		Island



3. Organización del proyecto

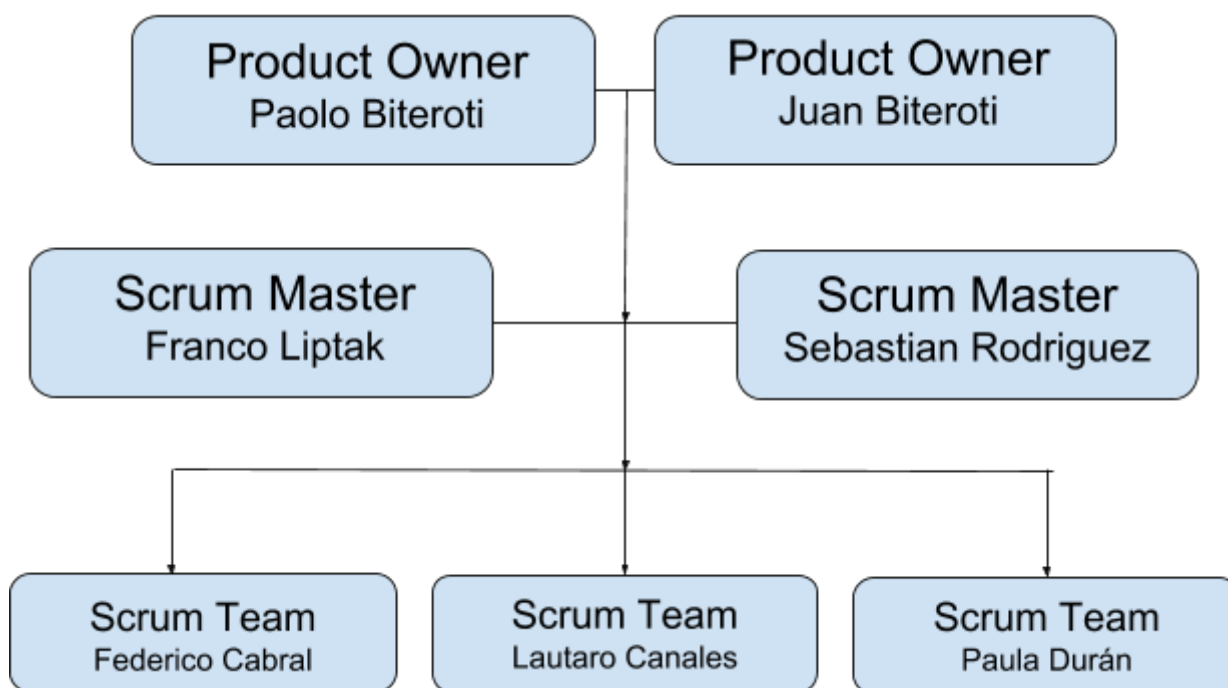
3.1. Interfaces externas

Los miembros del proyecto son los correspondientes a la metodología ágil a utilizar (Scrum).

Al utilizar la metodología ágil Scrum, el cliente es parte del equipo de desarrollo, por lo que la interacción entre las partes no se encuentra dificultada por barreras burocráticas.

En los puntos siguientes se explicará de mejor manera la estructura y la forma de trabajar con esta metodología.

3.2. Estructura interna



3.3. Roles y responsabilidades

Los roles y las responsabilidades de cada uno de los integrantes del proyecto están dados por la metodología ágil a utilizar (SCRUM).

En detalle:

Product Owner (Dueño del producto): Paolo y Juan.



Es quien representa al cliente (en este caso, los clientes mismos), usuarios de software y todas las partes interesadas en el producto. Es responsable de definir las prioridades del proyecto.

Scrum Master: Sebastian y Franco.

Es quien dirige el proyecto. Es responsabilidad del mismo garantizar la correcta aplicación de los principios de scrum. Debe incentivar y motivar al equipo, creando un clima de trabajo colaborativo, fomentar la autogestión del equipo e impedir la intervención de terceros en la gestión del equipo.

Scrum Team (Equipo): Federico Cabral, Lautaro Canales, Paula Durán.

Es el equipo de desarrolladores que debe llevar a cabo el desarrollo del producto.

4. Planes de administración del proceso

4.1. Plan inicial

4.1.1. Plan del personal

El personal a utilizar esta dado por los desarrolladores del proyecto, el scrum team. Los mismos se encargaran de diseñar, desarrollar, testear y la puesta en marcha del mismo.

El tiempo en que se precisa de ellos es el correspondiente a todo el desarrollo de la aplicación.

4.1.2. Plan de adquisición de recursos

No se requiere de pagos de licencias para el uso o adquisición del software utilizado para el desarrollo.

Se requiere de la compra de un dominio y el alquiler de un hosting para la aplicación web.

El dominio recomendado es unaventon.com.ar que se encuentra disponible para su utilización. La adquisición del mismo se realizará a través de la pagina <https://nic.ar/>, siguiendo los requisitos de la normativa vigente determinada por la página.

El costo del dominio es de \$270 para dar el alta del mismo, con una renovación anual de \$270.

Para el hosting se alquilara a la empresa <https://www.com.ar>, que ofrece un paquete intermedio a \$58 mensuales, sin límite de uso de disco ni de transferencia de datos.

4.1.3. Plan de entrenamiento del personal del Proyecto

Para poder llevar a cabo el proyecto, los desarrolladores deberán capacitarse en el lenguaje de programación Ruby, en el framework Rails, en el



uso de las gemas de Ruby, en el lenguaje de programación de etiquetado HTML, CSS, JavaScript, Bootstrap, entre otras.

Los IDEs elegidos para trabajar serán Visual Studio Code, y Brackets. También se debe capacitar a los desarrolladores con el uso de esos dos IDEs.

Para aprender Ruby se utilizará en mayor medida la documentación provista por <https://www.codecademy.com>, <https://www.ruby-lang.org/en/documentation/>, <https://medium.freecodecamp.org/learning-ruby-from-zero-to-hero-90ad4eccc82d>, <https://www.tutorialspoint.com/ruby/index.htm>, entre otros.

Para aprender el framework Rails de Ruby se utilizará entre tantos, <https://gorails.com>, <https://www.codecademy.com/learn/learn-rails>, <https://www.railstutorial.org/book>, <https://www.youtube.com/watch?v=GY7Ps8fqGdc>.

Las gemas de Ruby son varias, por lo que se considerará buscar las más conocidas y utilizadas de las cuales se encuentre documentación.

Para aprender HTML se considerará la página <https://www.w3schools.com/>, <https://www.codecademy.com/learn/learn-html>, <https://html.spec.whatwg.org/>.

En caso de necesitar documentación de Bootstrap se utilizará como referencia la página <https://getbootstrap.com>, y <https://www.w3schools.com/bootstrap4/default.asp>.

En caso de necesitar documentación de CSS se utilizará como referencia la página <https://www.w3schools.com/css/>.

En caso de necesitar documentación referida a JavaScript se utilizará la página <https://www.w3schools.com/js/default.asp>.

Para el IDE Brackets se puede buscar en caso de ser necesario en la documentación ofrecida por <http://css-snippets.com/brackets-course/>.

Para el IDE Visual Studio Code se puede ver los videos introductorios ofrecidos por <https://code.visualstudio.com/docs/introvideos/basics>.

4.2. Plan de trabajo

4.2.1. Principales actividades del proyecto

Las actividades principales a llevar a cabo para el desarrollo del sistema consisten en las básicas para el desarrollo de Software. Consta de:

- **Análisis:** Consiste en la obtención de todos los requerimientos del producto de software solicitado, y la entrega correspondiente de los documentos creados a partir de ellos.
- **Diseño:** Consiste en descomponer y organizar el sistema en elementos que puedan realizarse por separado. También consiste en el diseño de la estructura interna del programa, la definición de los algoritmos empleados y la organización del código. Entrega de los documentos obtenidos a partir de ello.
- **Implementación:** Programación de los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior.
- **Pruebas:** Testeo de la implementación desarrollada. Debe cumplir los requerimientos detallados en la parte de análisis.



- Mantenimiento: Consta en la puesta en marcha y mantenimiento posterior del sistema de información.

4.2.2. Asignación de esfuerzo

Actividad	Cantidad	Esfuerzo (hs) Unitario	Esfuerzo (hs) Subtotal
Análisis	3	20	60
Diseño	3	24	72
Implementación	3	120	360
Pruebas	3	40	120
Mantenimiento	3	72	216
TOTAL		276	828

4.2.3. Asignación de presupuesto

El presupuesto estará dado por el pago de las horas trabajadas a los integrantes del scrum team, y las compras de licencias y alquileres necesarios en un año para mantener la aplicación funcionando.

Se considera que el pago de una hora de trabajo por empleado debe contar con seguro médico, aportes, y el monto correspondiente al salario en mano. De esta forma podemos decir que por uso de conocimientos específicos, esfuerzo y trabajo dedicado, la hora de desarrollo de sistemas se pagará \$150.

Con los costos ya asignados hasta ahora, se procede a detallar el costo total del proyecto.

Elementos a Entregar	Cantidad	Precio Unitario	Precio Total
Alquiler anual de Dominio	1	\$ 270	\$270
Alquiler mensual hosting	12	\$58	\$696
Horas de trabajo	828	\$150	\$124200
TOTAL			\$125166

4.3. Plan de administración de riesgos

Se encuentra desarrollado de manera detallada en un posterior documento denominado "Anexo de Administración del Tratamiento de Riesgos".



5. Planes de procesos técnicos

5.1. Modelo de proceso

El modelo de proceso está definido por el modelo de proceso de la metodología ágil a utilizar denominada Scrum.

Scrum define un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.)

Scrum permite la creación de equipos auto-organizados impulsando la co-localización de todos los miembros del equipo.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan.

Este modelo de ciclo de vida es una manera de realizar proyectos de desarrollo de software de manera rápida, es una forma de desarrollar eficientemente ya que cada proceso que va avanzando o las ideas que se tengan son conocidas de una forma inmediata ya que se evalúan día a día para ver cómo se va evolucionando en dicho proyecto o resolver las situaciones que se presenten o puedan presentarse a futuro que impidan el progreso del proyecto, y esto forma lo que es la innovación constante.

Este ciclo es muy efectivo por su constante innovación que presenta al momento de desarrollar un proyecto y su forma de organización y control que se maneja retroalimentando la participación en el desarrollo del proyecto.

Pila de Producto:

La pila de producto es el corazón de Scrum. Es donde empieza todo. La Pila de Producto es, básicamente una lista priorizada de los requisitos que componen el sistema a desarrollar. Los requisitos serán escritos como Historias de Usuario.

La pila del producto es propiedad del Dueño del Producto, es decir el cliente final o su representante.

Planificación del Sprint:

El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto, pone nombre a la meta de la iteración (de manera que ayude a tomar decisiones durante su ejecución) y propone los requisitos más prioritarios a desarrollar en ella. El equipo examina la lista, pregunta al cliente las dudas que le surgen, y selecciona los objetivos/requisitos más prioritarios que se compromete a completar en la iteración. Para ello toma como medida el esfuerzo que representa la implementación y puesta a punto de cada requisito.

Utilizaremos la técnica Planning Poker para calcular el esfuerzo de cada historia de usuario.

Demo:

Al finalizar el Sprint se realiza la Demo, que consiste en una reunión formal donde el equipo presenta al cliente los requisitos completados en la iteración.

Durante la Demo el cliente puede ver de manera objetiva como han sido desarrollados los requisitos que proporciono, ver si se cumplen sus expectativas, entender más qué es lo que necesita y tomar mejores decisiones respecto al proyecto.

Los requisitos a mostrar serán las historias de usuario completadas, para que el cliente no se haga falsas expectativas y pueda tomar decisiones correctas y objetivas en función de la velocidad de desarrollo y el resultado realmente completado. Un requisito no completado quedará como requisito mas para la siguiente iteración.

Reunión de Retrospectiva:



Con el objetivo de mejorar de manera continua su productividad y la calidad del producto que está desarrollando, el equipo analiza cómo ha sido su manera de trabajar durante la iteración, por qué está consiguiendo o no los objetivos a los que se comprometió al inicio de la iteración y por qué el incremento de producto que acaba de demostrar al cliente era lo que él esperaba o no. El Equipo debe contestar las preguntas ¿Qué cosas han funcionado bien? ¿Cuáles hay que mejorar?

Scrum Diario:

Son reuniones que realiza el Equipo durante el Sprint, donde el foco es determinar el avance de las tareas y detectar problemas o bloqueos. Cada integrante debe contestar las preguntas ¿En qué trabajé ayer? ¿En qué voy a trabajar hoy? ¿Qué problemas impiden mi progreso?

5.2. Métodos, herramientas y técnicas

El lenguaje elegido es Ruby, con framework Rails. Es necesario el uso de Gemas de Ruby. Es factible la necesidad del uso de html, css y javascript para trabajar tanto Ruby como Rails.

El motor de la base de datos será definido por el uso de Gemas de Ruby, o compatibilidad de Rails con alguno en especial.

Para el desarrollo de las historias de usuario y la puesta en marcha de las mismas, además de la planificación y control de su producción, se utilizara la herramienta pivotal tracker (<https://www.pivotaltracker.com>) .

Para el trabajo conjunto de los desarrolladores y manejar el versionado del proyecto se utilizara el programa Git, con uso de GitHub, residiendo el proyecto en la pagina <https://github.com/fede38/Ingenieria-de-software-2-grupo-31>.

5.3. Plan de infraestructura

Ya se cuentan con los recursos necesarios para el desarrollo del proyecto.

5.4. Plan de aceptación del producto

Al utilizar la metodología ágil Scrum para el desarrollo del proyecto, la aceptación del mismo por parte del cliente está garantizada, ya que es uno de los miembros activos del proyecto.

6. Plan de procesos de apoyo

6.1. Plan de documentación

La documentación utilizada que forma parte del proyecto consiste en todos los entregables:

- SRS.
- Entrevistas.
- DER.
- Historias de Usuario (pivotal).
- PGP.
- Anexo de Riesgos.



-
- Pilas de Sprint.

7. Planes adicionales

Plan de mantenimiento: Se ofrecerá 3 meses de mantenimiento por parte de los desarrolladores. En caso de requerir mayor tiempo de mantenimiento se deberá volver a contratar a la empresa para el servicio.

Plan de copias de respaldo: Se harán copias de seguridad diariamente de la base de datos para garantizar que no se pierdan los datos en caso de eventualidades.

Plan de encriptación: Se encriptarán las contraseñas y el número de tarjeta de crédito para prevenir robo de los mismos.