

Universidad ORT Uruguay

Facultad de Ingeniería

Sistema de gestión de obras para una empresa constructora

Entregado como requisito para la obtención del título de Analista Programador

Gabriel Martínez – 188532

Tutores: Fernando Thul, Pablo Volpi

2015

Declaración de autoría

Yo, Gabriel Martínez, declaro que el trabajo que se presenta en esta obra es de mi propia mano.
Puedo asegurar que:

- La obra fue producida en su totalidad mientras realizaba el curso de proyecto integrador AP.
- Cuando he consultado el trabajo publicado por otros, lo he atribuido con claridad.
- Cuando he citado obras de otros, he indicado las fuentes. Con excepción de estas citas, la obra es enteramente mía.
- En la obra, he acusado recibo de las ayudas recibidas.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, he explicado claramente que fue contribuido por otros, y que fue contribuido por mí.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Firma del autor:

Gabriel Martínez

27/08/2015

Agradecimientos

En primer lugar, agradezco al cliente, Jorge Carámbula por poner a disposición su tiempo, asumiendo el rol, apoyando y guiando en todo momento hacia la mejor solución tanto académica como para el propio beneficio de la empresa.

A los tutores involucrados, Fernando Thul y Pablo Volpi, por sus aportes, valiosas críticas constructivas y su contribución al crecimiento del proyecto. Al profesor Raúl Gonzales por sus correcciones en la primera versión del anteproyecto en el curso de ingeniería de software. A Fernando Martínez y Florencia Polcaro por contribuir con su experiencia y comentarios para mejorar la presentación en la defensa del proyecto.

A los correctores Hugo Olivera y Santiago Fagnoni por sus aportes y recomendaciones durante la defensa.

Y por último a los profesores y coordinadores por su disposición a responder consultas en todo momento y por todo lo aprendido durante el transcurso de la carrera.

Abstract

La tecnología de la Información está cambiando la forma tradicional de hacer las cosas, las personas que trabajan en el gobierno, en empresas privadas, que dirigen personal o que trabajan como profesional en cualquier campo utilizan la TI cotidianamente mediante el uso de Internet. Esto brinda actualmente una cantidad ilimitada de recursos que pueden ser aplicados en diversas áreas del conocimiento para mejorar y perfeccionar paulatinamente los procesos para obtener resultados más eficaces, eficientes y en menor tiempo. Es por eso que la función de la TI en los procesos de las empresas y en especial las empresas constructoras se ha expandido grandemente.

Aplicando los conocimientos aprendidos durante el transcurso de la carrera este proyecto se enmarcará dentro de lo anteriormente mencionado, desarrollando un sistema de gestión de obras para una empresa constructora. Siendo el principal beneficiario IGAFAL S.A, pero a su vez manteniendo una fácil adaptación a nuevos requerimientos y permitiendo la reutilización a otras posibles entidades.

Como se mencionó este documento trata sobre el desarrollo de un sistema de gestión de obras para una empresa constructora y perfectamente el cual se podría utilizar para cualquier otra empresa constructora ya que no está desarrollada para una sola en particular.

Una de las principales motivaciones a desarrollar este proyecto se debe a que en el mercado para este tipo sistemas de gestión no se encuentran un gran número de herramientas y además, las herramientas actuales no engloban el conjunto de la obra sino tratan apartados concretos.

En definitiva, tomando todos los datos expuestos en los párrafos anteriores se llega a la conclusión de que hacen falta herramientas que gestionen las obras en su conjunto, permitan reducir costes, además faciliten el trabajo de los empleados donde se le brinde a los mismos una herramienta que pretende ser fácil, intuitiva y que cumpla todos los objetivos para los que es diseñada, consiguiendo así una gestión de obra a bajo coste. Es por ello que nace el sistema de gestión de obras de una constructora.

Palabras clave

Sistema de gestión, construcción, estimación, recursos, materiales, rubros, procedimientos constructivos, incidentes, nómica de contrataciones, obras, proyectos, sub contratos, costos indirectos, asesoramientos, proveedores, usuarios, cargos, subcontratistas, clientes, cálculos, gráficas, noticias, roles, parámetros, emails, sms, instrucciones, módulos, verificación en dos pasos.

Índice

Declaración de autoría.....	2
Agradecimientos	3
Abstract.....	4
Palabras clave	5
1. Anteproyecto	10
1.1. Introducción.....	10
1.1.1. Descripción de la empresa	10
1.1.2. Descripción del sistema a desarrollar	10
1.2. Análisis estratégico	12
1.2.1. Misión.....	12
1.2.2. Visión.....	12
1.3. Análisis FODA	13
1.3.1. Fortalezas	13
1.3.2. Oportunidades	13
1.3.3. Debilidades.....	13
1.3.4. Amenazas	14
1.4. Actores involucrados.....	14
1.5. Introducción de las herramientas a utilizar	15
1.6. Visión del producto	16
1.6.1. ¿Qué hace hoy el cliente?	16
1.6.2. ¿Cómo lo hace hoy el cliente?	16
1.6.3. ¿Cómo lo va a hacer con el nuevo sistema?	17
1.7. Objetivos	17
1.8. Stakeholders	18
1.9. Lista de necesidades	19
1.9.1. Lista de requerimientos	21
1.9.2. Requerimientos no funcionales:.....	24
1.9.3. Backlog de producto	25
1.10. Planificación	26
1.10.1. Inicio del proyecto sprint 1 y 2.....	27
1.10.2. Lista de tareas	28
1.10.3. Pila de producto general.....	37
1.11. Descripción del entorno.....	38

1.11.1. Diagrama de clases	38
1.11.2. Diagrama de componentes.....	40
1.12. Alcances y limitaciones	41
1.13. Estudio de alternativas	42
1.13.1. Alternativa física 1.....	44
1.13.2. Alternativa física 2.....	46
1.13.3. Alternativa física 3.....	48
1.13.4. Opción técnica 1.....	50
1.13.5. Opción técnica 2.....	51
1.13.6. Opción técnica 3.....	52
1.13.7. Selección de la alternativa	53
1.14. Análisis de riesgos	55
1.14.1. Tabla de riesgos	57
1.14.2. Plan de contingencia MMMR.....	58
1.15. Plan de Proyecto	60
1.15.1. Definición del Proceso	60
1.15.2. Incrementos o iteraciones definidas.....	61
1.15.3. Integrantes y roles	62
1.15.4. Plan de SQA.....	65
1.15.5. Estándares definidos y convenciones	65
1.15.6. Plan de testing.....	66
1.15.7. Plan de SCM	68
1.15.8. Plan de capacitación	69
1.15.9. Cronograma de trabajo y criticidad	69
1.15.10. Plan de instalación	73
1.15.11. Diagrama de despliegue.....	73
2. Desarrollo	74
2.1. Introducción.....	74
2.2. Sprint 1.....	74
2.2.1. Reuniones.....	74
2.2.2. Reunión retrospectiva.....	79
2.3. Sprint 2	80
2.3.1. Análisis del anteproyecto.....	80
2.4. Sprint 3	81

2.4.1. Especificación de requerimientos del sprint.....	81
2.4.2 Conclusiones del sprint	83
2.4.3. Reuniones.....	87
2.5. Sprint 4	88
2.5.1. Especificación de requerimientos del sprint.....	88
2.5.2 Conclusiones del sprint	90
2.5.3. Reuniones.....	93
2.5.4. Reunión de retrospectiva.....	97
2.6. Sprint 5	98
2.6.1. Especificación de requerimientos del sprint.....	98
2.6.2 Conclusiones del sprint	101
2.6.3. Reuniones.....	106
2.6.4. Reunión de retrospectiva.....	108
2.7. Sprint 6	109
2.7.1. Especificación de requerimientos del sprint.....	109
2.7.2 Conclusiones del sprint	111
2.7.3. Reuniones.....	113
2.7.4. Reunión de retrospectiva.....	114
2.8. Sprint 7	115
2.8.1. Especificación de requerimientos del sprint.....	115
2.8.2 Conclusiones del sprint	116
2.8.3. Reuniones.....	118
2.8.4. Reunión de retrospectiva.....	119
2.9. Sprint 8	120
2.9.1. Especificación de requerimientos del sprint.....	120
2.9.2 Conclusiones del sprint	121
2.9.3. Reuniones.....	125
2.9.4. Reunión de retrospectiva.....	126
2.10. Tabla de cumplimiento de requerimientos	127
2.11. Funcionalidades del producto, casos de uso	128
2.12. Otros requerimientos no funcionales	136
2.13. Requisitos para la interfaz externa	137
2.14. Diagramas	138
2.14.1. Evolución de las entidades del modelo.....	138

2.14.2. Secuencia	141
2.14.3. Arquitectura.....	142
2.15. Accesos web.....	145
2.16. Despliegue para cada producto entregable.....	146
2.17. Utilización de la nube de Azure	151
2.18. Aplicación web	156
2.19. Aplicación móvil adicional.....	167
2.19.1. Desarrollo.....	167
2.19.2. Sobre la herramienta	168
2.19.3. Resultado obtenido.....	169
2.20. Performance y usabilidad	175
2.21. Conclusiones finales.....	179
2.21.1. Introducción.....	179
2.21.2. Lecciones aprendidas	179
2.21.3. Posibles mejoras	181
2.21.4. Desarrollos a futuro	182
2.21.5. Conclusiones	182
Glosario	183
Lista de referencias bibliográficas.....	185
Bibliografía	186

1. Anteproyecto

1.1. Introducción

1.1.1. Descripción de la empresa

La empresa constructora IGAFAL S.A posee un servicio en el campo de la arquitectura e ingeniería donde se brinda servicios referentes a la realización de obras. Se busca la excelencia en su trabajo, la satisfacción de sus clientes y la seguridad de los trabajadores.

Toda la atención está dirigida a comprender las necesidades de los clientes y cumplir con sus expectativas de calidad, costo y plazo, desarrollándose prácticas comerciales y formas contractuales altamente flexibles para atender las necesidades de distintos tipos de proyectos y clientes.

La empresa tiene la versatilidad necesaria para adecuarse a distintos tipos de trabajo. Cada proyecto es estudiado hasta sus mínimos detalles cumpliendo la misión de ofrecer la mejor relación precio/calidad para los clientes.

1.1.2. Descripción del sistema a desarrollar

Como se mencionó para cada proyecto se presentan ofertas de precios para distintos rubros de un llamado a licitación pública. Estas obras pueden ser por ejemplo construcción de edificaciones, casas, apartamentos, suministro de electricidad, agua, gas, saneamiento, limpieza, obras viales, plazas entre otros.

Se solicita crear un sistema que controle los precios de los diferentes rubros, la gestión de los empleados de la empresa, los materiales y maquinaria a utilizar en cada proyecto. También se solicita mejorar la atención al cliente a través de una web donde se exponga y publicite todos los emprendimientos de la empresa, un módulo de contratación de servicios y distintos módulos privados para los empleados donde puedan ir ingresando sus tareas. Por ejemplo, el director de obra controlaría que para su proyecto asignado se están siguiendo las normas y las pautas establecidas en el contrato del proyecto, tipo, cantidades de materiales, ubicación etc.

Para estimar los precios para las licitaciones en la denominada “aperturas de precios” se necesitará que el sistema calcule desde los materiales y el personal hasta la maquinaria a utilizar, siendo algunos materiales a ejemplo cemento, hierro, cañerías de hormigón, pvc o fundición, maderas de encofrado, tablones para apuntalamiento, arena, pedregullo, materiales de relleno como balasto, materiales menores como clavos, alambre, etc. Las maquinarias pueden ser retroexcavadora, grúas, camiones, generadores, compresores, palas mecánicas, motoniveladoras para pavimento, etc.

Se tiene que tener en cuenta para el cálculo de precios las subcontrataciones a empresas terceras especializadas en algún rubro como por ejemplo para calles se usa el material

bituminoso - carpeta asfáltica, en lugar del hormigón, por lo general esto se subcontrata a empresas especializadas, porque es algo muy específico.

Como parte de la fachada “publicitaria” se espera que se puedan ofrecer vacantes de trabajo donde se pueda participar enviando un *curriculum* y ver información sobre el trabajo. Esta información contendrá tipo de cargo, área, actividad de la empresa, región (rural o metropolitana) y ciudad de residencia. A su vez se mostrarán los requisitos como, estudios mínimos, situación de estudios, carrera, experiencia mínima, conocimientos en computación, duración y jornada laboral.

Se creará además un historial llamado nómina de contrataciones públicas y privadas donde se mantendrá un registro de los clientes, descripción y ubicación de la obra.

Es importante destacar que el sistema tendrá dos secciones generales que englobarán lo que es presentación al cliente, vacantes disponibles, la fachada “publicitaria” de la empresa y otra sección que englobe todas las funciones privadas para los empleados de la misma.

Possiblemente a esto se le adicione un módulo de prevención de accidentes donde se le permitirá a un prevencionista realizar sus tareas de gestión y prevención de accidentes.

La principal ventaja a obtener del sistema en general es el cálculo del mejor precio para la licitación y mejorar la presentación de la empresa constructora al cliente por medio de una web interactiva y moderna.

1.2. Análisis estratégico

Como el cliente se desempeña en un entorno competitivo se realizó este análisis basado en los factores críticos de éxito. Se incluyó la misión y visión de la actividad del cliente, así como el análisis FODA (Fortalezas, Oportunidades, Debilidades y Amenazas) del proyecto a desarrollar.

1.2.1. Misión

- Unificación de la estrategia de trabajo
 - Conseguir que esta herramienta mejore los procesos tanto humanos como técnicos y una todos los procesos relacionados con las constructoras
- Mejorar la primera estimación de precio para un proyecto
 - Una de las partes más primordiales de este proyecto es la de idear un proceso que le proporcione al cliente una estimación rápida y sencilla para un proyecto, que a su vez pueda retroalimentarse con la información obtenida mejorando cada vez más la estimación inicial.
- Modernización de procesos de gestión de empresa y publicitarios
 - Como parte adicional a este proyecto se buscará que este sistema mejore la imagen empresarial del cliente y le brinde una herramienta que le permita estar un paso más adelante de su competencia.

1.2.2. Visión

- Mantener el prestigio de nuestro cliente como el de cualquier otra empresa constructora a lo largo del tiempo.
- Evolucionar de forma tal que los cambios a la empresa sean fácilmente reflejables en el sistema
- Ofrecer lo último en
- tecnologías de *Windows ASP.NET* para las empresas constructoras

1.3. Análisis FODA

De acuerdo con las características del equipo de proyecto y las necesidades, entorno y forma de trabajo del cliente, se identificaron los siguientes aspectos.

1.3.1. Fortalezas

- Se cuenta con el tiempo necesario para cumplir con la lista de tareas definida más adelante, entregando los sprint a tiempo.
- Se cuenta con los conocimientos necesarios para realizar todo lo propuesto y el correcto cumplimiento del proyecto.
- Se dispone de un tutor de la Universidad ORT, el cual se encuentra dispuesto a indicar, sugerir y guiar al proyecto para asegurar que la calidad sea la adecuada.
- Se dispone de todo lo referente a hardware y software para el análisis, diseño, desarrollo y testeo del producto mencionado.
- El desarrollador del proyecto cuanta con una actitud positiva y proactiva lo cual hace que el trabajo y aprendizaje sea algo deseable.

1.3.2. Oportunidades

- La oportunidad estratégica ya que actualmente no existe ningún sistema que posea las cualidades anteriormente descriptas.
- La oportunidad de mejorar la forma en que se realiza una primera estimación de precios para un proyecto.
- La oportunidad de ofrecerle al cliente una herramienta que fue ideada según sus necesidades y elaborada con dedicación y análisis meticuloso de los procesos de la empresa.
- Esta es una oportunidad de elaborar no solo un proyecto que beneficie al cliente sino también la realización personal donde se obtengan nuevos conocimientos y experiencia sobre la tecnología emprendida.

1.3.3. Debilidades

- Podría ocurrir un margen de error ya que se calcularán estimaciones de precios.
- La falta de experiencia desarrollando planificaciones para proyectos.
- La falta de conocimientos totales sobre las herramientas de desarrollo.
- La falta de conocimientos informáticos de cliente.
- El no entendimiento de todos los aspectos de la arquitectura podría causar diferencias a la hora del relevamiento de necesidades del cliente.

1.3.4. Amenazas

- Segmento en el mercado contraído.
- Difícil de monetizar.
- Inexistencia de competencia, la competencia daría pautas y metas para mejorar.
- La idea del proyecto podría ser eventualmente copiada.
- Al ser un proyecto académico se tienen que brindar los accesos a los servidores.
- Cambios abruptos en la lista de precios oficial.
- Cambios abruptos de los requerimientos.

1.4. Actores involucrados

En la sección privada laboral o intranet denominada “sistema de gestión integrada” se espera un sistema de *logueo* de usuarios donde se permita ver:

Las asignaciones a tareas, el historial, porcentaje de antigüedad en la empresa, grupo de trabajo, proyecto, tiempo estimado para realizar el proyecto, infracciones cometidas, entre otros. Estas variaran según el rol de usuario y su rol en la empresa.

Algunos roles del personal de la empresa del sistema de gestión integrada serán:

- Director de obra - Control general de una o varias obras
- Ingeniero residente (o arquitecto) - está siempre en la obra
- Topógrafo/ing. Agrimensor - realiza metrajes, relevamientos y replanteos topográficos
- Ayudante o dibujante
- Capataz general
- Administrativo - por ejemplo, paga los jornales y toda otra tarea
- Cantonero - Se encarga de distribuir herramientas de trabajo y materiales
- Encargado de cuadrillas
- Maquinistas, carpinteros, herreros, albañiles, peones

Para la parte del sistema de presentación al cliente habrá un administrador y los usuarios generales que se registrarán para recibir noticias y promociones, también serán capaces de ingresar su *curriculum* y participar en las vacantes disponibles.

1.5. Introducción de las herramientas a utilizar

El cliente especificó que todo debe ser accesible mediante web y que los servidores a utilizar pueden ser rentados a conveniencia. Por lo tanto, se optó por utilizar tecnologías que dimos en el curso y a mi parecer son suficientes para realizar este sistema.

Para la parte de presentación se utilizarán *HTML*, *CSS* y *JavaScript* que fue lo que se especificó al decir “accesible mediante web” este será generado dinámicamente por el *framework* ASP.NET, el dominio será desarrollado en *C#* y la persistencia en *SQL Server*.

Para la parte móvil posiblemente se desarrollarán aplicaciones para los principales sistemas operativos móviles utilizando Phonegap.

De forma general usar *HTML5* junto a los estilos de Bootstrap sumado a las posibles aplicaciones móviles en Phonegap beneficiará al proyecto con una nueva estructura que permitirá por ejemplo definir por separado el encabezado, la barra de navegación, las secciones de la página web, los textos del sitio, los diálogos y el pie de página sin importar en qué sistema operativo se esté. Esta nueva estructura permitirá desarrollar toda la interfaz del sistema con gran facilidad, ofreciendo además la posibilidad de obtener un código más limpio y fácil de depurar, incluso con más facilidad que los códigos de los estándares y metodologías anteriores.

Los patrones de diseño serán determinados más adelante cuando se haga un estudio de la complejidad, tiempo y costos que requiere este sistema.



Fig. 1 Ilustración HTML, JS y CSS

1.6. Visión del producto

¿Qué beneficio le trae al cliente el nuevo sistema?

El principal beneficio del producto es lograr satisfacer la necesidad de calcular una estimación inicial de precio para un proyecto arquitectónico, teniendo en cuenta los diversos rubros involucrados en la obra.

Con este proyecto intentaremos llegar al valor más cercano del “metro cuadrado de obra”. Dicha expresión es comúnmente utilizada ante una entrevista inicial con un cliente de la empresa constructora.

1.6.1. ¿Qué hace hoy el cliente?

IGAFAL S.A brinda proyectos y construcciones a clientes o inversores nacionales o internacionales dentro del ramo de la arquitectura. Dichos proyectos pueden ser tanto públicos como privados siguiendo los más altos estándares constructivos, preocupándose por su imagen empresarial y sus ganancias intentando mejorar paulatinamente los beneficios obtenidos y la seguridad de sus empleados.

1.6.2. ¿Cómo lo hace hoy el cliente?

Generalmente mi cliente como muchas otras empresas constructoras se basan en la experiencia y antecedentes para estimar dichos precios, estos pueden tener un amplio margen de error dando lugar a pérdidas económicas no deseadas.

Normalmente IGAFAL S.A estima los precios a “mano” utilizando documentos brindados por el ministerio de transporte y obras públicas tales como la lista oficial de precios de materiales, herramientas, saneamiento, pintura, madera, seguridad, hierro, eléctrica y albañilería junto a sus respectivos proveedores. Además de otro documento publicado por el ministerio de trabajo y seguridad social referente a los laudos vigentes, como lo son los salarios para los obreros jornaleros, obreros mensuales, administrativos, generales, oficiales, técnicos entre otros.

1.6.3. ¿Cómo lo va a hacer con el nuevo sistema?

Dicho todo lo expresado anteriormente es necesario aplicar nuevos estilos en la gestión de las obras, una manera nueva de tratar todo el conjunto del proyecto de una obra de manera diferente a la actual. El porqué de esto es que actualmente hay mucha falta de control en el progreso de una obra, tales como muchos costos que no se especifican, muchas desviaciones en el presupuesto, lo que conlleva que en situaciones como la mencionada, las constructoras se descapitalicen.

Tras presenciar esta realidad se llegó a la conclusión de que hay que cambiar procedimientos y mejorarlos para que todo pueda seguir funcionando, pudiendo conseguir que las empresas en particular IGAFAL S.A ahorren costes y es por ello que con este sistema a desarrollar se pretende subsanar todos estos problemas y simultáneamente dar una nueva visión al proceso de construcción de cualquier obra. A lo largo del contenido de este anteproyecto se irá viendo paso a paso como llegar a crear un sistema intuitivo, rápido, sencillo y moderno que ayude a las empresas constructoras seguir adelante en momentos tanto como previos a realizar un proyecto o a lo largo de la gestión de los mismos.

1.7. Objetivos

Los objetivos que plantea la constructora suponen varias mejoras y disminuciones en los diferentes rubros de la misma.

Por el área de la fachada publicitaria se pretende lograr:

- OFP1: Aumentar la contratación de recursos humanos en un 20%
- OFP2: Aumentar la publicidad de la empresa alcanzando al menos un 8% de la población
- OFP3: Generar ganancias por parte de los nuevos recursos humanos en un 30%
- OFP4: Disminuir el 50% de los gastos en prácticas manuales de contratación de personal
- OFP5: Aumentar la contratación de jóvenes en un 5%

Por el área del sistema de gestión integrada se pretende lograr:

- OSGI1: Disminuir los precios presentados en la licitación en 5%
- OSGI2: Mejorar la puntualidad y la asistencia de los empleados en un 40%
- OSGI3: Aumentar la reutilización de los materiales almacenados en 20%
- OSGI4: Mejorar los tiempos de entrega promedio en 10%
- OSGI5: Evitar la totalidad de las perdidas por hurtos en materiales

1.8. Stakeholders

Stakeholder sería la parte interesada. Se puede definir como cualquier persona o entidad que es afectada o concernida por las actividades o la marcha de una organización.

Todos estos obtienen beneficios o sufren daños como resultado de las acciones de la empresa.

Nuestros *stakeholders* internos son:

Dueños

Constructora IGAFAL S.A: Accionistas

Empleados

Director de obra - Control general de una o varias obras
Ingeniero residente (o arquitecto) - está siempre en la obra
Gerentes
Ing. Agrimensor/topógrafo
Ayudante o dibujante
Capataz general
Administrativo
Cantonero
Encargado de cuadrillas
Maquinistas, carpinteros, herreros, albañiles, peones
Escribano, contador, abogado
Prevencionista

Y los externos son:

Externos

Asociaciones empresariales, industriales, o profesionales
Clientes: los inversores de la empresa IGAFAL S.A
Competidores: Steeler, Teyma, Osu
Comunidades donde la empresa tiene operaciones: asociaciones vecinales, comunas, caminos vecinales
Gobierno nacional
Gobiernos departamentales y municipios
Inversores: Sociedades anónimas nacionales o internacionales
Medios de comunicación, inmobiliarias, agencias de bienes raíces: ACSA
ONGs: Si es una obra social puede participar
Proveedores/vendedores a la empresa: Hormigones del Uruguay, Ancap (Portland), CIR (Metalurgica), Barraca Central (Ladrillos bloques), subcontratistas varios
Sindicatos: Sunca

Fig. 2 Tablas de stakeholders internos y externos

1.9. Lista de necesidades

Se utilizará una serie de entrevistas guiadas por el equipo de relaciones donde se reunirán con los usuarios para entender mejor sus necesidades, motivaciones, cultura laboral y varios aspectos adicionales. Las mismas están definidas en el siguiente fragmento del cronograma de Gantt.

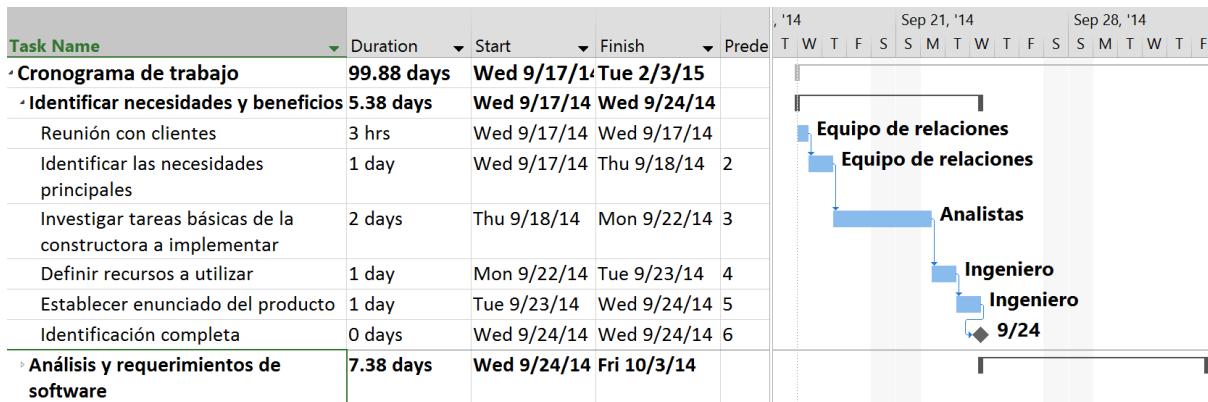


Fig. 3 Diagrama de Gantt (2014) - Identificar necesidades y beneficios

Es importante destacar que estas son las primeras entrevistas correspondientes al anteproyecto de ingeniería de software del 2014 más adelante durante el proyecto de fin de carrera habrá una lista de tareas y entregas intermedias (*sprints*) donde se irán refinando las necesidades y corroborando las encontradas previamente. En la página siguiente se puede observar el nuevo diagrama.

En un principio las necesidades básicas son:

- Mejorar seguridad de la empresa y sus trabajadores
- Mejorar la estimación de precios de los proyectos
- Gestionar materiales, maquinaria, empleados de la empresa
- Mejorar la atención a los clientes con una fachada publicitaria
- Crear un módulo de contratación de servicios
- Crear un módulo de control de empleados
- Controlar normas y pautas para los proyectos
- Generar una estimación de precios para la “apertura de precios”
- Crear Historial o nómina de contrataciones públicas y privadas
- Dividir el sistema en dos áreas “Sistema de gestión integrada” y “fachada publicitaria”
- Mejorar la interactividad, modernizando el sistema.

Una vez identificadas las necesidades básicas correspondientes al 2014 se procedió a crear otro diagrama Gantt donde se implementó el marco de desarrollo ágil denominado *Scrum* en el cual se volvió a reunir, agregando nuevos requerimientos y verificando que estas anteriores seguían vigentes en el 2015.

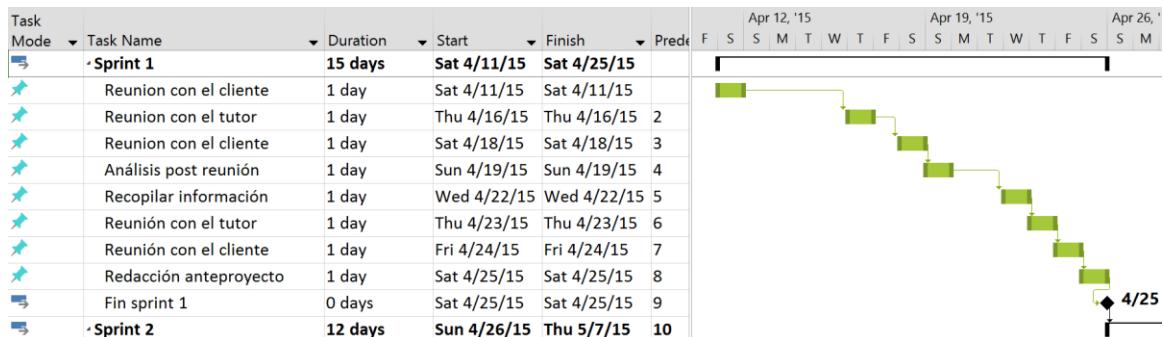


Fig. 4 Diagrama de Gantt - Identificar necesidades y beneficios

La guía que deberá seguir todo integrante que conforme el equipo de relaciones para las entrevistas con el cliente es la Guía Avanzada de gestión de Requisitos [4] publicada por Incibe (Instituto Nacional de Ciberseguridad), donde se debe leer desde el capítulo Buenas Prácticas ubicado en la página 18 en adelante.

1.9.1. Lista de requerimientos

A continuación, definiremos los requerimientos funcionales, estos son declaraciones de los servicios que debe proporcionar este sistema, la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.

Y por otro lado definiremos los no funcionales, que son restricciones de los servicios o funciones ofrecidos por el sistema.

Requerimientos funcionales:

- RF 1: Página de inicio de sesión
 - El sistema debe tener una página inicial para el sistema de gestión integrada donde se puedan *loguear* todo tipo de usuarios.
- RF2: Acceso a los módulos del sistema
 - El sistema se debe poder visualizar mediante módulos del sistema y estructuras básicas necesarias.
- RF3: Dos módulos fachada y sistema de gestión integrada
 - El sistema debe de estar claramente dividido en dos grandes módulos que engloben y categoricen las funcionalidades.
- RF4: Gestionar al personal
 - El sistema debe permitir dar de alta, baja y modificar todo el personal estando organizados según su rol, con determinados permisos y diferentes tareas. A si según su rol podrán acceder a diferentes áreas del sistema entre otros.
- RF5: Gestionar los roles
 - El sistema debe permitir dar de alta, baja y modificar roles para asignárselo al personal.
- RF6: Gestionar los clientes
 - El sistema debe permitir dar de alta, baja y modificar cliente de la empresa constructora y asignarles proyectos.
- RF7: Gestionar proyectos
 - El sistema debe permitir dar de alta, baja y modificar proyectos a los cuales se le asignaran obras.
- RF8: Gestionar obras
 - El sistema debe permitir dar de alta, baja y modificar obras a las cuales se le asignarán componentes que podrán ser materiales y/o personal. Estas podrán ser reutilizadas para los proyectos
- RF9: Gestionar los recursos y proveedores.
 - El sistema debe permitir dar de alta, baja y modificar todos los materiales como "recursos" ya sean maquinaria, herramientas, equipos informáticos, documentos que posea la empresa deben de estar ingresados y organizados. Cada uno de ellos tendrá asociado un proveedor.
Cabe aclarar que se considera al personal de una obra como un componente calculable.

- RF10: Gestionar componentes
 - El sistema debe permitir dar de alta, baja y modificar componentes que serán formados por materiales, estos componentes pueden a su vez formar parte de otros componentes.
- RF11: Gestionar estados para las obras
 - El sistema debe categorizar las obras según su estado donde se podrá distinguir entre obras que estén en planeamiento, desarrollo, finalizadas y en asesoramiento. Esta última se utilizará cuando un cliente solicite asesoramiento para una obra, completando los datos de la solicitud y donde al finalizar el arquitecto le enviará para esa obra en estado de asesoramiento una estimación de costo.
- RF12: Gestionar relaciones entre componentes
 - El sistema debe permitir como se mencionó en el RF10 que los componentes estén conformados por otros componentes siendo el tipo de componente más básico los materiales de la lista de precios oficial del ministerio de transporte y obras públicas.
- RF13: Informes sobre el estado de una obra en particular
 - El sistema debe permitir generar un informe de todos los materiales, maquinaria, herramientas, trabajadores que participen en una obra. Estos deben ser monitoreados para poder generar un informe de estado sobre la obra.
- RF14: Informes de todas las contrataciones públicas/privadas
 - El sistema debe permitir generar un informe de todas las contrataciones públicas y privadas de la empresa deben ser registradas y ordenadas cronológicamente con los respectivos recursos utilizados para la misma. Por lo tanto, una contratación deberá expandirse con todos los recursos y detalles.
- RF15: Ingreso de las pautas y criterios para cada proyecto
 - El sistema debe permitir que para cada proyecto exista un monitoreo sencillo donde se ingrese que pautas y criterios que se requieren.
- RF16: Informes de precios estimados en base al proyecto
 - El sistema debe permitir que en base al proyecto cuya obra u obras están conformada por componentes se calcule una estimación de precio total. Teniendo en cuenta el beneficio a obtener y otras variables.
- RF17: Exponer cargos disponibles en la empresa.
 - El sistema debe mostrar cuando existan cargos disponibles en la empresa, estos se podrán exponer en la fachada publicitaria, permitiendo que los interesados envíen sus formularios y posteriormente si cumplen con los requerimientos agendar una entrevista.
- RF18: Prevenir y gestionar accidentes
 - El sistema deberá permitir que los empleados de la constructora tengan la seguridad de si al sufrir un accidente lo podrán reportar y será revisado por un prevencionista el cual tomará las medidas necesarias.

- RF19: Calcular la paramétrica para una obra que no sea de este año
 - El sistema debe permitir que para una obra que no sea de este año se pueda actualizar su precio mediante las paramétricas del ministerio de transporte y obras públicas. Esto ayudara a mantener que las obras del sistema sean reutilizables.
- RF20: Sección de noticias o novedades sobre la empresa
 - El sistema debe exponer todas las noticias y/o novedades de la empresa las cuales serán dirigidas al público en general.
- RF21: Reportar un accidente
 - El sistema debe permitirle a cualquier miembro del personal a reportar un accidente, el cual será gestionado por un prevencionista para su posterior relevamiento.
- RF22: Tiempo sin accidentes
 - El sistema debe mostrar el tiempo sin accidentes en la fachada publicitaria para mejorar la imagen empresarial. Donde en caso de que exista un accidente empezar de cero y mostrar qué medidas se tomaran para subsanar y futura prevención.
- RF23: Cronograma financiero
 - El sistema debe especificar que para cada fase de la obra una previsión de los gastos. Donde al final se compara el precio estimado inicial con el total del cronograma financiero.
- RF24: Cronograma físico
 - El sistema debe permitir que para cada fase de la obra se registre cuando se deberían completar la construcción de los rubros.
- RF25: Cronograma de obtención de equipos y materiales
 - El sistema debe permitir que para cada fase de la obra se registre que materiales se usaron y que equipos se deberían obtener.
- RF26: Instrucciones para el proceso de estimación, desarrollo de obras y otros
 - El sistema debe mostrar todos los procesos que posee el sistema en instrucciones que expliquen los pasos que se deben seguir para desarrollarlos además de definir y explicar brevemente que realiza cada funcionalidad.
- RF27: Asesoramiento para clientes
 - El sistema debe permitir que un cliente pida un asesoramiento, especifique que requerimientos quiere de una obra y reciba una estimación de costos.

1.9.2. Requerimientos no funcionales:

- RNF1: El sistema debe seguir principios de usabilidad y accesibilidad.
- RNF2: El sistema debe ser adaptable fácilmente a nuevas funcionalidades debido a la naturaleza cambiante del proyecto, utilizando técnicas y patrones arquitectónicos que sean escalables, manteniendo una alta cohesión y bajo acoplamiento.
- RNF3: El sistema debe ser portable a otras tecnologías y plataformas si es necesario.
- RNF4: El sistema debe ser multiplataforma, *Browser, IOS* y *Android*
- RNF5: El sistema debe ser estable en todo momento, por lo cual se deberá comprobar con alguna herramienta de estadísticas que el sistema llega a un mínimo del 95% del tiempo de ejecución sin caídas por cada mes. Esto se comprobará en el punto 2.20. de performance y usabilidad.
- RNF6: La interfaz gráfica debe ser accesible en un tiempo menor a 5 segundos, no los procesos ni consultas complejas, sino aquellas pantallas que le brinden al usuario una navegación fluida por el sistema. Esto se comprobará en el punto 2.20. de performance y usabilidad.
- RNF7: Evitar ataques típicos al sistema como *DDoS*, *hijacking* al servidor, adivinación de contraseñas por fuerza bruta, robo de datos no encriptados, etc.

1.9.3. Backlog de producto

<i>Nombre de requerimiento</i>	<i>Nº de requerimiento</i>
Página de inicio de sesión	RF1
Acceso a los módulos del sistema	RF2
Dos módulos fachada y sistema de gestión integrada	RF3
Gestionar al personal	RF4
Gestionar los roles	RF5
Gestionar los clientes	RF6
Gestionar los proyectos	RF7
Gestionar las obras	RF8
Gestionar recursos y proveedores	RF9
Gestionar componentes	RF10
Gestionar estados para las obras	RF11
Gestionar relaciones entre componentes	RF12
Informes sobre el estado de una obra en particular	RF13
Informes de todas las contrataciones públicas/privadas	RF14
Ingreso de las pautas y criterios para cada proyecto	RF15
Informes de precios estimados en base al proyecto	RF16
Exponer cargos disponibles en la empresa.	RF17
Prevenir y gestionar accidentes	RF18
Calcular la paramétrica para una obra que no sea de este año	RF19
Sección de noticias o novedades sobre la empresa	RF20
Reportar un accidente	RF21
Tiempo sin accidentes	RF22
Cronograma financiero	RF23
Cronograma físico	RF24
Cronograma de obtención de equipos y materiales	RF25
Instrucciones para el proceso de estimación, desarrollo de obras y otros	RF26
Asesoramiento para clientes	RF27

Fig. 5 Tabla de backlog de producto

1.10. Planificación

Para el proyecto se decidió utilizar una adaptación de la metodología ágil de desarrollo denominada *Scrum* que se usa para minimizar los riesgos durante la realización de un proyecto, pero de manera colaborativa.

Entre las ventajas se encuentran la productividad, calidad y que se realiza un seguimiento diario de los avances del proyecto, logrando que los integrantes estén unidos, comunicados y que el cliente vaya viendo los avances.

Esto se basa en que un proyecto se ejecuta en bloques temporales (iteraciones de un mes natural y hasta de dos semanas) llamados “*Sprints*”. Cada iteración o sprint tiene que proporcionar un resultado completo, un incremento de producto que sea potencialmente entregable, de manera que cuando el cliente lo solicite sólo sea necesario un esfuerzo mínimo para que el producto esté disponible para ser utilizado.

Según lo mencionado por Sommerville [1], aunque el enfoque sea *Scrum*, su enfoque está en la administración iterativa del desarrollo, y no en enfoques técnicos específicos para la ingeniería de *software* ágil.

Este proceso no prescribe el uso de prácticas de programación, como la programación en pares y el desarrollo de primera prueba. Por lo tanto, puede usarse con enfoques ágiles más técnicos, como XP, para ofrecer al proyecto un marco administrativo.

Existen tres fases con *scrum*. La primera es la planeación del bosquejo, donde se establecen los objetivos generales del proyecto y el diseño de la arquitectura de software. A esto le sigue una serie de ciclos sprint, donde cada ciclo desarrolla un incremento del sistema. Finalmente, la fase de cierre del proyecto concluye el proyecto, completa la documentación requerida, como los marcos de ayuda del sistema y los manuales del usuario, y valora las lecciones aprendidas en el proyecto.

En la siguiente página se definirán los *sprints* necesarios en la que se valora el trabajo que se va a realizar, se seleccionan las particularidades por desarrollar y se utilizará para implementar el software. Al final de un sprint, la funcionalidad completa se entregará al cliente en forma de “producto entregable”.

1.10.1. Inicio del proyecto sprint 1 y 2

En esta etapa del proyecto se fueron analizando los requerimientos junto con reuniones con el cliente en donde se investigó sobre el sistema a desarrollar. Se partió como base el anteproyecto entregado en el curso de Ingeniería de software y se lo fue modificando acorde a los nuevos requerimientos y metodologías de trabajo.

- En el sprint 1:
 - Se definió el proyecto
 - Descripción la empresa y del sistema a desarrollar
 - Se estableció la visión del producto
 - Se definió los objetivos a cumplir
 - Se definió los *stakeholders*
- En el sprint 2:
 - Lista de necesidades
 - Lista de requerimientos y *backlog* de producto
 - Planificación del proyecto
 - Pila de producto
 - Descripción del entorno y selección de herramientas
 - Descripción del entorno
 - Diagrama conceptual de clases
 - Diagrama de componentes
 - Cronograma de trabajo y criticidad
 - Integrantes y roles
 - Selección de herramientas a utilizar
 - Análisis de riesgos
 - Plan de SQA
 - Plan de SCM
 - Plan de capacitación

1.10.2. Lista de tareas

1.10.2.1. Sprint 1

<i>Id</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
<i>S1T1</i>	7	Reunión con el cliente	Relevamiento inicial	11/04 - 11/04	2hs
<i>S1T2</i>	8	Reunión con el tutor	Presentación del proyecto y análisis	16/04 — 16/04	2hs
<i>S1T3</i>	8	Reunión con el cliente	En base a las observaciones del tutor, se relevara las necesidades del cliente más incisivamente	18/04 — 18/04	3hs
<i>S1T4</i>	5	Análisis post reunión	Necesidades del cliente y pila de producto	19/04 — 19/04	4hs
<i>S1T5</i>	8	Recopilar información	Visita a la intendencia y cámara de la construcción en búsqueda de información necesaria	22/04 — 22/04	5hs
<i>S1T6</i>	6	Reunión con el tutor	Ánalisis del proyecto en base a la información conseguida	23/04 — 23/04	2hs
<i>S1T7</i>	5	Reunión con el cliente	Vía online presentarle lo hablado en la reunión con el tutor	24/04 — 24/04 25/04	1hs
<i>S1T8</i>	8	Redacción anteproyecto	Modificación y redacción	— 25/04	6hs
<i>Análisis</i>			<i>Entrega pendiente de sprint 2</i>	---	25hs

Fig. 6 Tabla de tareas del sprint 1

Diagrama de Gannt correspondiente al sprint 1

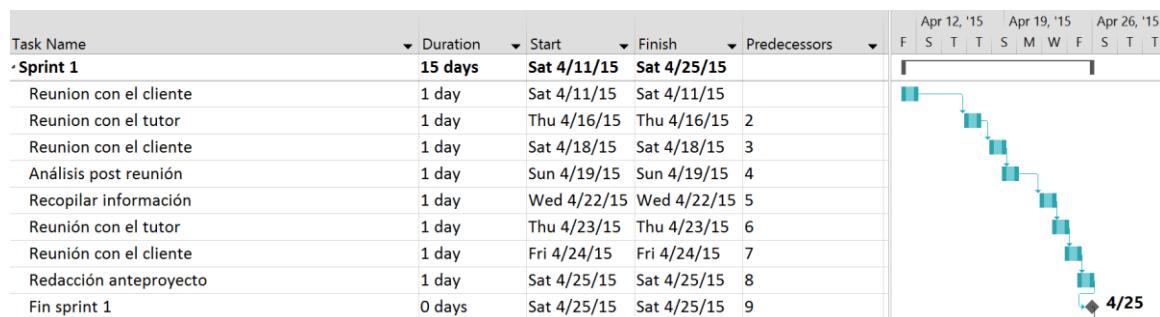


Fig. 7 Diagrama de Gantt del sprint 1

1.10.2.2. Sprint 2

<i>Id</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
S2T1	7	Redacción anteproyecto	Modificación y redacción	26/04 – 26/04	8hs
S2T2	9	Anteproyecto pilas y tareas	Pila de producto, lista de tareas	27/04 – 29/04	36hs
S2T3	9	Reunión con el tutor	Análisis del proyecto	30/04 – 30/04	2hs
S2T4	6	Diagrama de clases	Diagrama de clases UML	01/05 – 01/05	6hs
S2T5	5	Diagrama MER	Diagrama para la persistencia de los datos	02/05 – 02/05	6hs
S2T6	6	Actualizar Gant	Actualiza diagrama de Gant según las fechas de las tareas	03/05 – 05/05	15hs
S2T7	7	Acoplar diagramas	Acoplar diagramas a la documentación	06/05 – 06/05	6hs
S2T8	5	Correcciones generales	Todo debe estar listo y verificado para la entrega del sprint 2	07/05 – 07/05	3hs
<i>Diseño</i>			<i>Entrega de sprint 1 + 2</i>	<i>07/05</i>	<i>82hs</i>

Fig. 8 Tabla de tareas del sprint 2

Diagrama de Gannt correspondiente al sprint 2

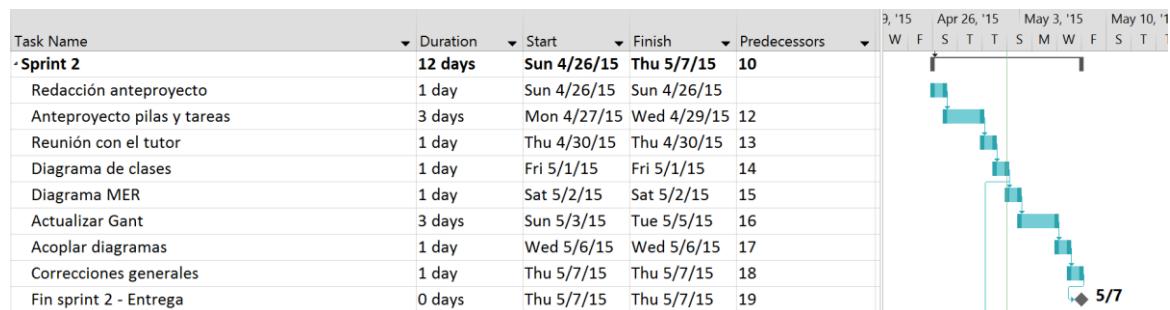


Fig. 9 Diagrama de Gantt del sprint 2

1.10.2.3. Sprint 3

<i>Id</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
S3T1	9	Plataforma tecnológica	Setear un ambiente de testeo	01/05 – 03/05	8hs
S3T2	7	Primeras pruebas	NVC, Entity Framework test, Hola mundo!	04/05 – 04/05	6hs
S3T3	7	Configuración de entorno	DNS, FTP, <i>backups</i> , versionado	05/05 – 05/05	6hs
S3T4	8	Crear estructura de contenido	Crear el “esqueleto de las clases” del proyecto	06/05 – 06/05	4hs
S3T5	9	Crear persistencia	Crear la persistencia a partir de Entity Framework	07/05 – 07/05	4hs
S3T6	3	Probar funcionamiento	Testear clases ingresando datos genéricos	08/05 – 08/05	6hs
S3T7	7	Ingresar datos existentes	A partir de la lista de precios ingresar datos ya sea consumiendo un web service o parseando el Excel	09/05 – 10/05	16hs
<i>Sistema de gestión int.</i>				<i>Entrega pendiente de sprint 3</i>	---
					50hs

Fig. 10 Tabla de tareas del sprint 3

Diagrama de Gannt correspondiente al sprint 3



Fig. 11 Diagrama de Gannt del sprint 3

1.10.2.4. Sprint 4

<i>Id.</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
S4T1	5	Ingresar datos relaciones	Dado lo anterior crear relaciones entre las entidades de prueba	11/05 –	5hs
S4T2	6	Componentes estructura	Crear la estructura necesaria para modelar la idea de los “componentes”	11/05 12/05 –	5hs
S4T3	4	Modificación de estilos	En el sistema de gestión integrado mejorar estilos	13/05 – 15/05 16/05	14hs
S4T4	7	Roles	Roles para el personal	– 16/05	5hs
S4T5	6	Autentificación del personal	Crear sesión, logueo roles y todo lo necesario	17/05 – 18/05	10hs
S4T6	8	Permisos para los módulos	Permisos, tabla de permisos según módulos	19/05 – 25/05	25hs
S4T7	6	Análisis para el cálculo de precios	Según los precios ingresados de los componentes y materiales, analizar soluciones posibles para calcular	26/05 – 01/06	35hs
S4T8	8	Testeo y correcciones generales	Todo debe estar listo y testeado para la entrega del sprint 3	02/06 – 02/06	8hs
S4T9	7	Últimas modificaciones y documentación	Documentar todo lo sucedido en el transcurso de este sprint	03/05 – 04/06	7HS
<i>Sistema de gestión int.</i>			<i>Entrega sprint 3 + 4</i>	<i>04/06</i>	<i>114hs</i>

Fig. 12 Tabla de tareas del sprint 4

Diagrama de Gantt correspondiente al sprint 4

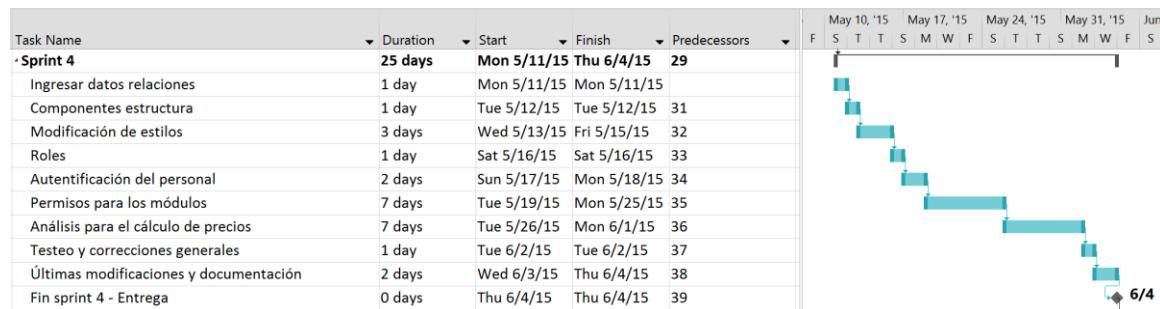


Fig. 13 Diagrama de Gantt del sprint 4

1.10.2.5. Sprint 5

<i>Id</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
S5T1	9	Crear primera versión de cálculo de precios	Dado el análisis final del sprint 3, crear versión experimental para la estimación de precios	05/06 – 10/06	30hs
S5T2	5	Verificar versión experimental con el cliente	Mostrarle la versión experimental al cliente y obtener comentarios	11/06 – 11/06	6hs
S5T3	7	Modificaciones para el cálculo de precios	En base a los comentarios, modificar o mejorar la versión experimental	12/06 – 15/06 16/06	30hs
S5T4	6	Ingresar datos de prueba	Ingresar datos de prueba para el cálculo de precios	16/06 – 16/06	4hs
S5T5	7	Modificar estilo de la versión experimental	Hacer que esta sección sea concuerde con el sistema de gestión integrada	17/06 – 18/06	10hs
S5T6	7	Persistir cálculos	Todos los cálculos tienen que persistirse y estar relacionados con un proyecto	19/06 – 20/06	8hs
<i>Sistema de gestión int.</i>			<i>Entrega pendiente de sprint 5</i>	---	88hs

Fig. 14 Tabla de tareas del sprint 5

Diagrama de Gannt correspondiente al sprint 5



Fig. 15 Tabla de tareas del sprint 5

1.10.2.6. Sprint 6

<i>Id.</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
S6T1	9	Crear interfaz para selección de tipo de proyecto	Crear una interfaz interactiva que permita al usuario seleccionar de los tipos de obra y crear su proyecto a medida	21/06 – 25/06	18hs
S6T2	5	Modificación de componentes para obra	Cuando el usuario selecciona un tipo de obra y lo asigna a su proyecto se le debe permitir modificar los componentes	26/06 – 30/06	25hs
S6T3	7	Persistir la modificación	Persistir la modificación como un nuevo tipo de obra, para su posterior reutilización	01/07 – 03/07	10hs
S6T4	8	Suggestions al escribir materiales o componentes	Cuando un usuario escriba los componentes para calcular los precios estos deben ser sugeridos fácilmente	04/07 – 05/07	8hs
S6T5	7	AJAX al agregar componentes o modificar	Cada vez que se agregue componentes estos deben aparecer instantáneamente	06/07 – 10/07	25hs
S6T6	6	Testeo y correcciones generales	Todo debe estar listo y testeado para la entrega del sprint 4	11/07 – 14/07	20hs
S6T7	7	Últimas modificaciones y documentación	Documentar todo lo sucedido en el transcurso de este sprint	15/07 – 16/07	8hs
<i>Sistema de gestión int.</i>			<i>Entrega sprint 5 + 6</i>	<i>16/07</i>	<i>114hs</i>

Fig. 16 Tabla de tareas del sprint 6

Diagrama de Gannt correspondiente al sprint 6



Fig. 17 Diagrama de Gantt del sprint 6

1.10.2.7. Sprint 7

<i>Id</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
S7T1	9	Diseñar fachada publicitaria	Diseñar fachada publicitaria de forma moderna y que siga las expectativas del cliente	16/07 – 17/07	8hs
S7T2	4	Búsqueda de herramientas adecuadas	Buscar librerías y herramientas que le aporten modernismo a la fachada publicitaria	18/07 – 18/07	3hs
S7T3	8	Implementar primer prototipo	Implementar el primer prototipo de la fachada publicitaria que sea compatible con el sistema de gestión integrada Todo lo que pueda ser dinámico y que aporte para la reutilización del sistema se debe considerar	19/07 – 20/07	4hs
S7T4	5	Analizar que puede ser dinámico en la fachada	Todo lo que pueda ser dinámico y que aporte para la reutilización del sistema se debe considerar	21/07 – 21/07	5hs
S7T5	8	Implementar dinamismos	Dado el análisis anterior crear toda la estructura necesaria	22/07 – 23/07	8hs
S7T6	7	Crear sección de contratación	Crear la sección que permite contratar nuevo personal con toda la información sobre la vacante	24/07 – 25/07	8hs
S7T7	6	Permitir postulantes	Crear la estructura necesaria para que un postulante tome una vacante	26/07 – 28/07	9hs
S7T8	5	Cerrar vacantes limitadas	Al llegar a la cantidad de postulantes de una vacante, cerrar la vacante	29/07 – 29/07	4hs
S7T9	8	Crear nómina de contrataciones	Tomando todos los datos del backend “sistema de gestión integrado” crear la nómina de contrataciones público privadas	30/07 – 31/07	8hs
S7T10	6	Tiempo sin accidentes	Crear el primer prototipo de tiempo sin accidentes, solo lo visual, en sprint 8 se mejora	01/08 – 02/08	8hs
S7T11	6	Mostrar progreso de completitud de proyectos	Crear el primer prototipo de una barra que muestre la completitud para cada proyecto de la nómina	03/08 – 03/08	6hs
<i>Fachada publicitaria</i>			<i>Entrega pendiente de sprint 7</i>	---	71hs

Fig. 18 Tabla de tareas del sprint 7

Diagrama de Gannt correspondiente al sprint 7

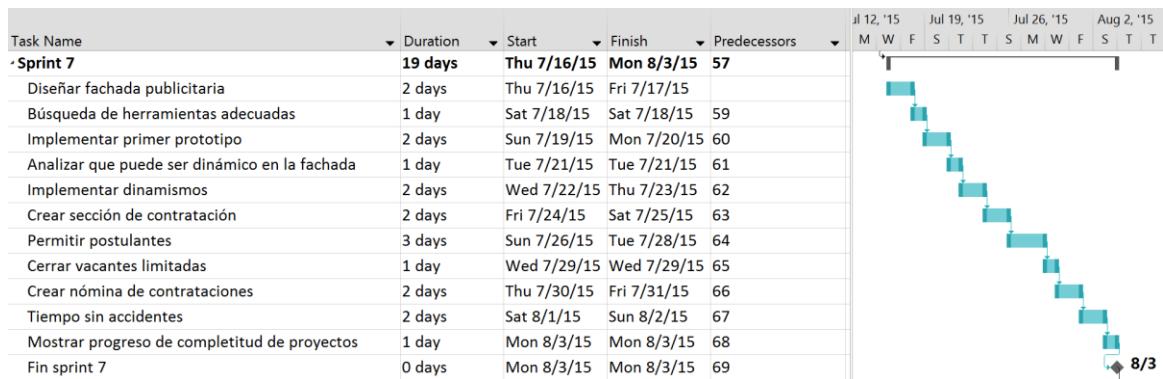


Fig. 19 Diagrama de Gannt del sprint 7

1.10.2.8. Sprint 8

<i>Id</i>	<i>Pri.</i>	<i>Nombre</i>	<i>Observación</i>	<i>Fecha</i>	<i>Est.</i>
S8T1	9	Diseñar prevención de accidentes	Diseñar un módulo prevencionista que ayude a fomentar la imagen de la empresa	04/08 – 04/08	8hs
S8T2	7	Implementar módulo prevencionista	Dado el análisis anterior crear el módulo prevencionista	05/08 – 07/08	10hs
S8T3	6	Compatibilizar módulo prevencionista	Hacer que la fachada y el módulo prevencionista interactúen	08/08 – 08/08	4hs
S8T4	8	Analizar fases para una obra	Analizar la forma más coherente de implementar las fases para una obra	09/08 – 09/08	4hs
S8T5	6	Implementar fases para una obra	Dado el análisis anterior crear la estructura necesaria para implementar las fases para una obra	10/08 – 10/08	6hs
S8T6	6	Cronograma financiero	Para cada fase de la obra, la previsión de los gastos. Al final se compara lo que compara el precio estimado inicial con el total del cronograma financiero	11/08 – 15/08	20hs
S8T7	7	Cronograma físico	Para cada fase de la obra registrar cuando se debería completar la construcción de los componentes	16/08 – 17/08	10hs
S8T8	4	Cronograma de obtención de equipos y materiales	Para cada fase de la obra registrar que materiales y equipos debería obtener	18/08 – 19/08	10hs
S8T9	5	Crear informe de completitud de proyectos	Dado los anteriores cronogramas crear un informe con posibles graficas que muestren la completitud del proyecto	20/08 – 22/08	18hs
S8T10	7	Testeo y correcciones generales	Todo debe estar listo y testeado para la entrega del sprint 8	23/08 – 24/08	50hs
S8T11	8	Últimas modificaciones y documentación	Documentar todo lo sucedido en el transcurso de este sprint	25/08 – 27/08	20hs

Sistema de gestión int. *Entrega de sprint 7 + 8* *27/8* *160hs*

Fig. 20 Tabla de tareas del sprint 8

Diagrama de Gannt correspondiente al sprint 8

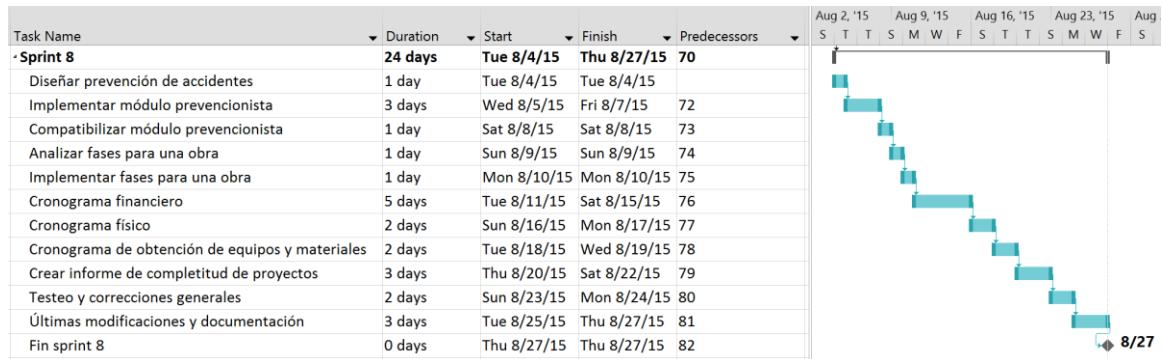


Fig. 21 Diagrama de Gantt del sprint 8

1.10.3. Pila de producto general

<i>Id</i>	<i>Prioridad</i>	<i>Descripción</i>	<i>Estimación</i>
P1	9	Plataforma tecnológica	20hs
P2	7	Gestión de materiales	35hs
P3	7	Gestión de personal	35hs
P4	7	Gestión de obra	65hs
P5	7	Gestión de componentes	65hs
P6	5	Gestión de incidentes	20hs
P7	5	Almacenamiento de materiales	6hs
P8	5	Asignación de materiales a obra	6hs
P9	5	Asignación de personal a obra	6hs
P10	5	Asignación de obra a cliente	6hs
P11	6	Estado de obra, alertas, incidentes	20hs
P12	6	Fases de una obra	10hs
P13	3	Vencimiento de alquileres de maquinaria	8hs
P14	7	Roles del personal	6hs
P15	7	Informe de completitud de proyectos, graficas	45hs
P16	6	Fachada publicitaria	30hs
P17	5	Ofrecer vacantes disponibles	10hs
P18	7	Postulantes a vacantes	10hs
P19	6	Fases de entrevistas a postulante	5hs
P20	5	Nómina de contrataciones publico/privadas	20hs
P21	7	Fases de un incidente, mejora del proceso	20hs
P22	4	Imagen empresarial tiempo sin acc. en la empresa	4hs
P23	10	Estimación de precio para una obra	260hs
P24	8	Cronograma financiero	20hs
P25	7	Cronograma físico	10hs
P26	4	Cronograma de obtención de equipos y materiales	10hs

Fig. 22 Pila de producto general

1.11. Descripción del entorno

En esta sección se adjuntarán algunos diseños tempranos sobre el entorno del proyecto. Los cuales sirven como representación de una posible solución para el problema planteado.

1.11.1. Diagrama de clases

Como se explicará en el siguiente capítulo para desarrollar el sistema se optó por utilizar Entity Framework. Un primer diseño del diagrama de clases para la opción “*model first*” de dicho patrón será adjuntado en la siguiente página.

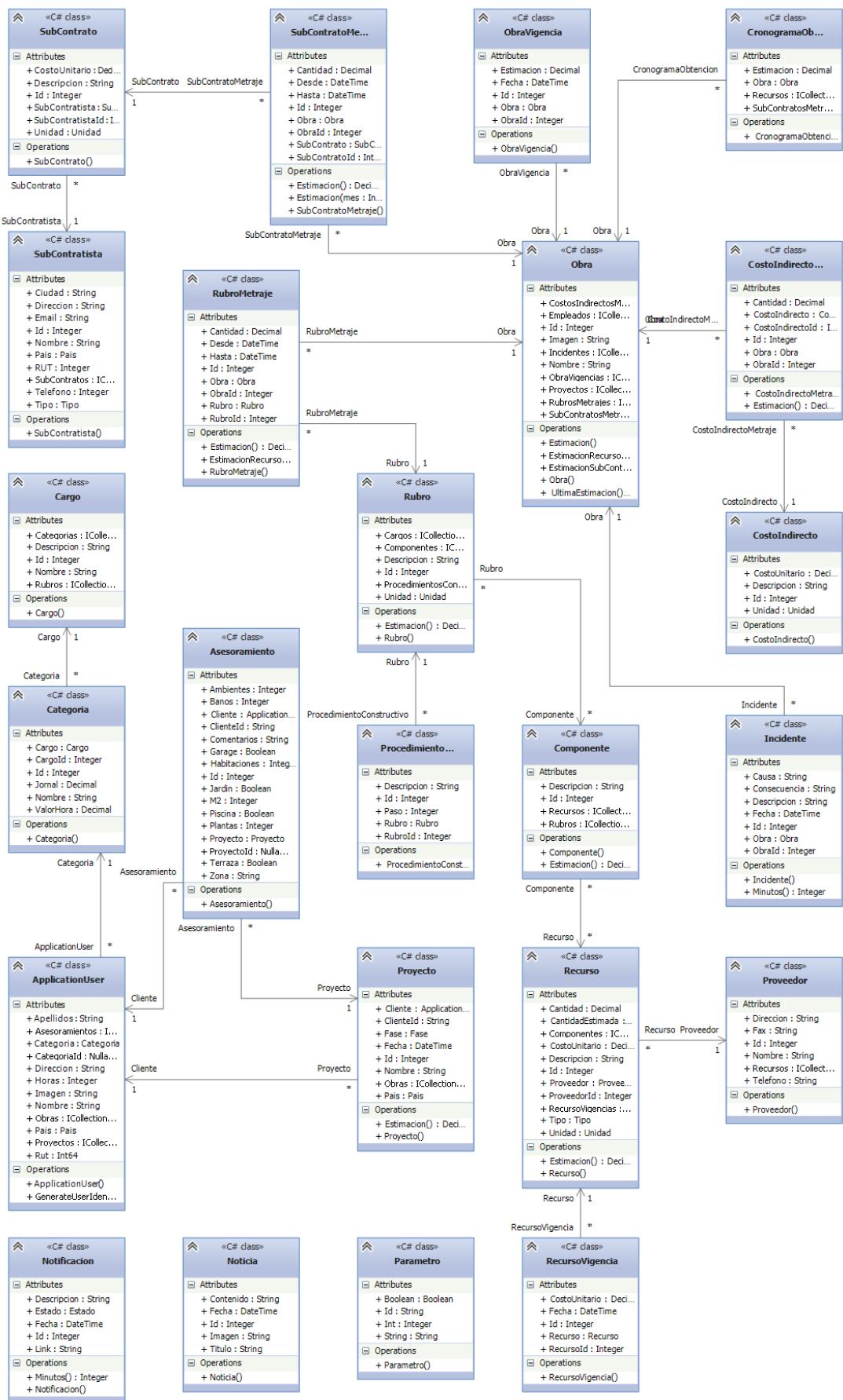


Fig. 23 Diagrama de clases

En este primer diagrama se puede observar las relaciones entre las entidades más importantes del sistema. Las cuales le dan al cliente el mayor beneficio y por donde se optó por comenzar con la codificación.

Con esta estructura de sistema donde los componentes, materiales y rubros están interrelacionados entre sí y donde los componentes se pueden formar por sí mismos se le brinda a un proyecto que puede estar conformado por varias obras la facilidad de cambiar el “contenido” de los rubros de una obra sin alterar todo el proyecto ni tampoco generar la necesidad de adentrarse a cambiar material por material.

Se espera que el cambio de componentes y la personalización de una obra pueda ser fácilmente realizado frente a un cliente en una entrevista por parte de un arquitecto mientras que se da una estimación de presupuesto más acertada según las exigencias que se vallan dando a lo largo del relevamiento.

1.11.2. Diagrama de componentes

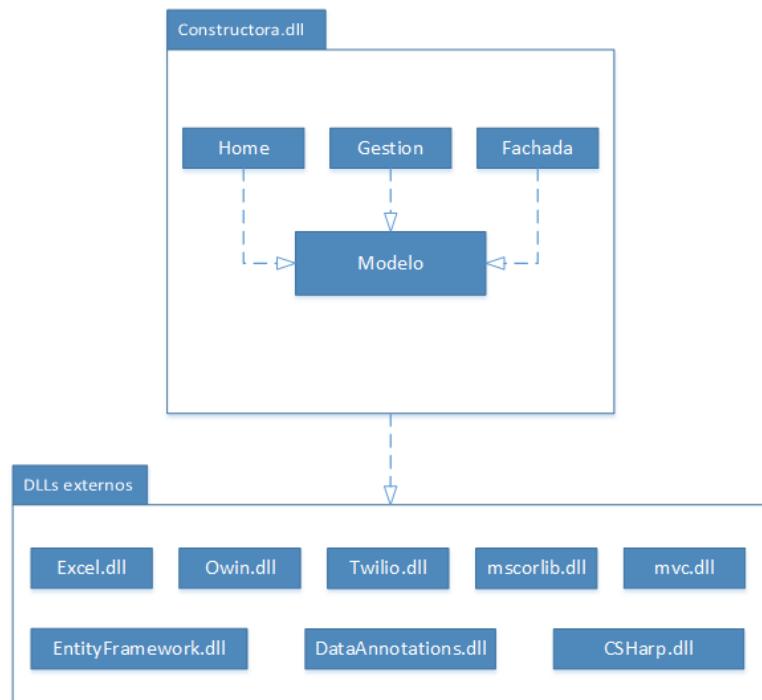


Fig. 24 Diagrama de componentes

1.12. Alcances y limitaciones

El área de alcance del proyecto se limita a las operaciones de gestión y cálculos estimativos de obras descriptas en el planteamiento del problema.

La aplicación, en especial el sistema de gestión integrada, podría contener módulos destinados a las estadísticas, datos de los empleados y procesos de la empresa, sin embargo, debido a que el plazo de entrega del proyecto es de diecisésis semanas y que en este ambiente dichas funcionalidades no son vitales para que un proyecto arquitectónico tenga éxito, es que se limita el alcance del Proyecto al área inicialmente descripta.

Como se ha expuesto en el análisis estratégico, donde se puede ver que el mundo de la construcción aún sigue siendo muy manual, que numerosos procesos pueden ser mejorados y llevados a cabo con una aplicación. Es un área que aún no está muy desarrollada y que se puede mejorar mucho, para ello es que se desarrolle un sistema que en un principio cumpla con lo más vital de un proyecto, la estimación de precios inicial y la gestión de obras.

Se concluyó de las reuniones con el cliente que, dentro de las operaciones de gestión y cálculo de precios para obras, existen otros procesos que pueden ser facilitados, pero ampliar el alcance en este sentido implicaría un riesgo de no completar el proyecto a tiempo o con las entregas de cada sprint.

Es en ese último sentido que también se limita el alcance del módulo de fachada publicitaria a la lista de necesidades descriptas en el punto 1.9 manteniendo abierta la posibilidad que en el futuro se pueda ampliar agregar funcionalidades y pueda ser integrado a los módulos existentes.

1.13. Estudio de alternativas

Posteriormente de definir la problemática y establecer las causas que ameritan de un nuevo sistema, es pertinente realizar un estudio de alternativas para determinar la infraestructura tecnológica y la capacidad técnica que implica la implementación del sistema en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la empresa constructora. Este análisis permitió determinar las posibilidades de diseñar el sistema propuesto y su puesta en marcha.

Analizando la factibilidad técnica se evaluó que el equipo y software están disponibles y se pueden desarrollar ya que el equipo dispone de los conocimientos básicos del *framework .NET* y del IDE Visual Studio 2013. Por lo tanto, se considera que los integrantes mencionados en el punto 1.15.3 poseen la experiencia técnica requerida para diseñar, implementar, operar y mantener el sistema propuesto. Si el personal no tuviese esta experiencia, puede entrenársele o emplearse nuevos que la tengan.

En el caso de que se requiera entrenamiento del personal se asignara un analista con experiencia como tutor a aquellos que requieran adquirir conocimientos sobre la *IDE Visual Studio 2013* y el entorno gráfico nuevo (respecto a versiones anteriores).

En este caso no se tendrá problemas relacionados con la experiencia ni con la contratación de personal que opere el equipo y herramientas mencionados a continuación, ya que este es un proyecto académico.

El presupuesto del proyecto cubre los costos y se plantearán tres alternativas de soluciones físicas ordenadas por costos de implementación y dos opciones técnicas referentes a la arquitectura de los componentes del sistema.

Esta división entre las alternativas y opciones sirve para independizar la parte de arquitectura de servidores con el modelo de componentes del sistema permitiendo que se elija la mejor dupla alternativa/opción para satisfacer las necesidades del cliente.

Se tuvo en cuenta que sin importar que alternativa se escoja esta será totalmente compatible con cualquier opción posible gracias a que se usa en todas ASP.NET.

Asimismo, ASP.NET sirve tanto para *Webs* sencillas como para grandes aplicaciones. Sin olvidar que la orientación a objetos y la naturaleza compilada permiten que se haga uso de herramientas de creación de *Webs* que facilitarán mucho la tarea de programación. Estas herramientas permiten hacer *Webs* sencillas y de bajas prestaciones en un tiempo moderado, así como llevar el mantenimiento de grandes aplicaciones de forma más sencilla.

Se tendrá la mayor velocidad, mayor potencia, mayor seguridad, mayor facilidad de mantenimiento y herramientas de trabajo que Microsoft junto a ASP.NET pueden ofrecer.

En resumen, las principales características de las tres alternativas físicas son:

1. Alternativa física 1
 - a. Alto costo
 - b. Alta escalabilidad
 - c. Alto performance
 - d. Independiente de terceros
2. Alternativa física 2
 - a. Costo medio
 - b. Alta escalabilidad
 - c. Medio-alto performance
 - d. Dependiente de terceros
3. Alternativa física 3
 - a. Costo bajo
 - b. Baja escalabilidad
 - c. Medio-bajo *performace*
 - d. Independiente de terceros

Y las principales características de las dos opciones técnicas son:

1. Opción técnica 1
 - a. *Testable*
 - b. Extensible
 - c. URLs semánticas
 - d. HTML limpio
 - e. *Routing*
 - f. *Open source*
2. Opción técnica 2
 - a. *ViewState*
 - b. Controles de servidor ricos
 - c. Desarrollo rápido
 - d. Baja curva de aprendizaje
3. Opción técnica 3
 - a. Simplicidad
 - b. Transaccional
 - c. Desarrollo rápido
 - d. Baja curva de aprendizaje
 - e. Experiencia con la herramienta

1.13.1. Alternativa física 1

Como primera alternativa se consideró utilizar un servidor “independiente” el cual podría generar un alto costo, pero una alta escalabilidad y performance.

Arquitectura: La aplicación sería instalada sobre un servidor en la empresa constructora, donde se ubicaría también la base de datos.

- **Ventajas:** crear el proyecto en un entorno propio. Requiere una inversión que no conlleva limitaciones, estas las impone el mismo proyecto. Se puede aumentar con la escalabilidad de servicios. Estar en un entorno propio y seguro permite estar tranquilo de la negligencia de los demás.
- **Desventajas:** la criticidad del proyecto conllevará una alta inversión, puede llegar a pasar que se caiga el servicio para evitar esto surge la necesidad de un buen equipo, con el mejor software, una gestión técnica, representa costos más elevados.

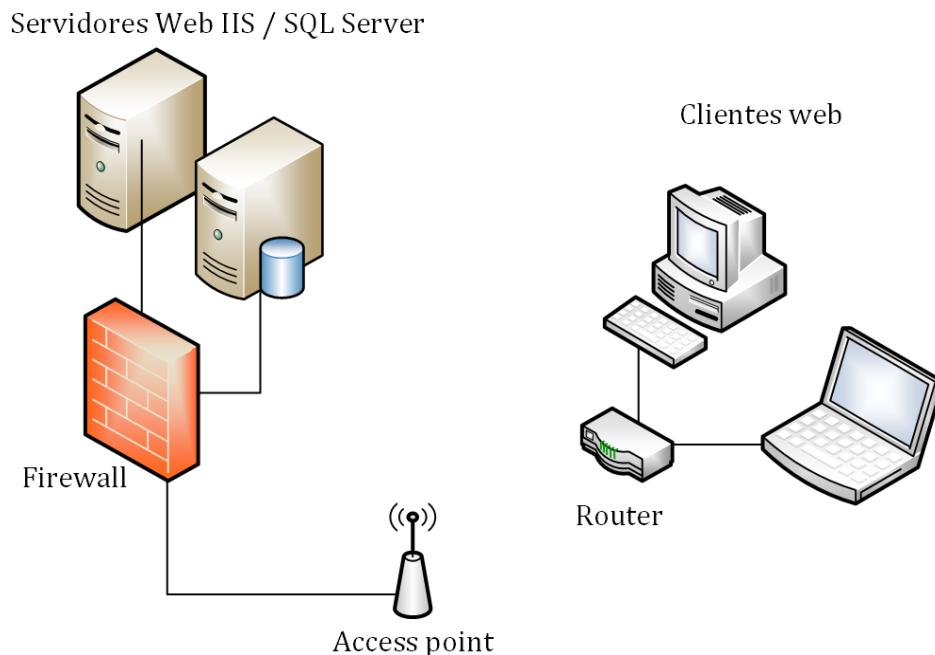


Fig. 25 Diagrama de alternativa física 1

Equipo o producto	Cantidad	Especificación
Computadoras	2	Intel Xeon E5-2600 8 Core, RAM 6 Gb, HDD 6TB
Backup	1*	Amazon Glacier
Sistema operativos	1*	Microsoft Server 2012 R2
Firewall	1*	IPCop
Gestor de base de datos	1*	SQL Server 2012
Cable de red	24	RJ-45 Apantallado
Switch	1	Linksys SE3024
IDE	1*	Visual Studio 2013 Ultimate
Conexión	1	Internet corporativo ANTEL
Sistema de climatización	1	12.000 BTU

*Licencias empresariales

Fig. 26 Tabla de alternativa física 1

1.13.1.1. Factibilidad operativa

Mediante esta alternativa la factibilidad operativa permite predecir que se podrá poner en marcha el sistema propuesto, aprovechando los beneficios que ofrece, a todos los usuarios involucrados con el mismo, ya sean los que interactúan en forma directa con este como los empleados de la empresa constructora, como también aquellos que reciben información producida por el sistema. Por otra parte, el correcto funcionamiento del sistema en cuestión, siempre estará sujeto a la capacidad de los empleados encargados de dicha tarea.

1.13.1.2. Factibilidad técnica

Se posee las habilidades de desarrollo, los conocimientos tecnológicos, recursos de hardware y software necesario para desarrollar el sistema propuesto. Los cuales en este caso son brindados por la Universidad ORT Finalmente, el tiempo que demandaría el desarrollo de esta alternativa estaría dentro de los plazos determinados por la universidad para la realización del proyecto.

1.13.1.3. Factibilidad legal

Esta alternativa es factible legalmente debido a que la mayoría de las herramientas utilizadas a pesar de no ser libres se cuenta con las licencias legales dentro del marco académico para desarrollar la aplicación. En caso de que el cliente pretenda obtener un mantenimiento y avances del proyecto fuera del plazo establecido, se le brindaría la opción para comprar dichas licencias junto con una nueva presupuestación.

Por otro lado, también se cumplirá con todas las normas que impone el software libre ya que como todo proyecto utilizará librerías que pueden llegar a ser *open source*.

1.13.1.4. Factibilidad Económica

Esta alternativa es factible económicamente ya que, el costo de análisis y desarrollo es costeable por la empresa ya que se trata de un proyecto de final de carrera, los costos de actualización de la infraestructura tecnológica de la empresa constructora donde se va a utilizar el sistema son nulos ya que no se brindará soporte posterior a las etapas del proyecto. Y los recursos existentes son suficientes para el correcto funcionamiento del producto y los costos de instalación.

Como se mencionó anteriormente en el estudio de factibilidad técnica, se cuenta con las herramientas necesarias para la puesta en marcha del sistema, por lo cual el desarrollo de la propuesta no requerirá de una inversión inicial.

1.13.2. Alternativa física 2

Como segunda alternativa se consideró utilizar un servidor de Microsoft llamado Azure, el cual es ampliamente configurable, con un costo moderado que permitiría estar en un punto medio entre las otras dos alternativas

Arquitectura: El sistema estaría basado en la arquitectura de Azure utilizando IIS y SQLServer en la nube.

- **Ventajas:** presencia en Internet a buen precio y sin inversión en hardware y software. Siendo el proveedor (Microsoft) la misma empresa que crea la tecnología a implementar. Se puede garantizar una máxima compatibilidad entre los distintos componentes del proyecto.
- **Desventajas:** poder compartir recursos abarata el precio, conlleva la limitación, disminución de la estabilidad y rendimiento de la web ya que el producto se encuentra alojado en el mismo servidor y con el mismo hardware que otros sistemas. Además, se puede ver afectado por las acciones de los otros sitios que están alojados en el servidor, como, por ejemplo, otro usuario haga spam, o sea víctima de un ataque de spam: la IP que compartida se listará en las *blacklists* o listas negras siendo imposible el funcionamiento del correo. Para evitar esto se utilizará una IP que no sea compartida.

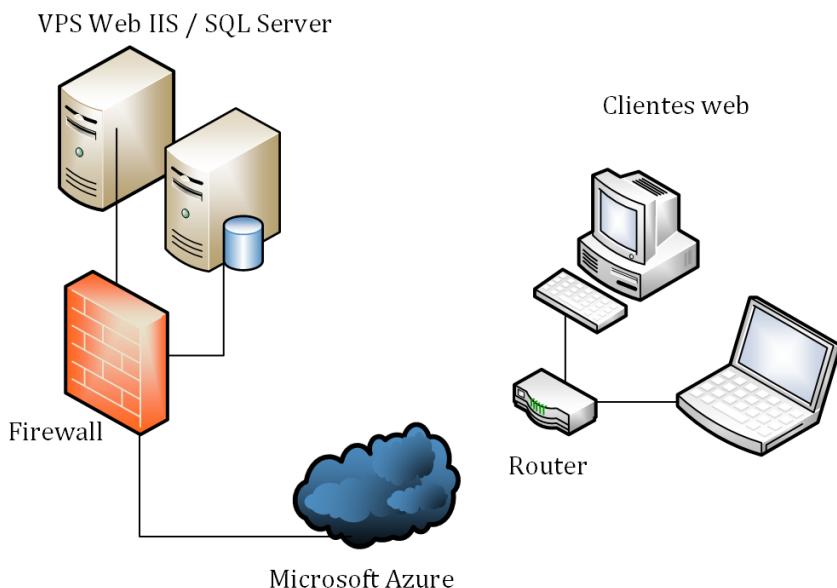


Fig. 27 Diagrama de alternativa física 2

Equipo o producto	Cantidad	Especificación
Computadoras de testeo	1	AMD FX-9590, RAM 4 Gb
Backup	1*	Azure backups
Sistema operativos	1*	IIS
Hosting	1*	Azure
IDE	1*	Visual Studio 2013 Ultimate
Conexión	1	Internet fibra óptica ANTEL
DNS	1	CloudFlare

*Licencias empresariales

Fig. 28 Tabla de alternativa física 2

1.13.2.1. Factibilidad operativa

Para esta alternativa la factibilidad operativa perímete predecir que se podrá en marcha el sistema propuesto, se aprovechará los beneficios como la plataforma lista para hacer un *deploy* sobre la nube de Microsoft, el sistema de *backups*, la base de datos relacional, entre otros.

Se aprovechará además la total compatibilidad que ofrece Visual Studio 2013 con Azure permitiéndonos desarrollar la aplicación en menos tiempo y más eficientemente.

1.13.2.2. Factibilidad técnica

Esta alternativa la podemos considerar como factible técnicamente porque, al analizar los requerimientos de la empresa, se estimó que los recursos de hardware, software y recursos humanos son adecuados para el correcto funcionamiento del producto. Asimismo, los integrantes del grupo de desarrollo poseen los conocimientos tecnológicos, recursos de hardware, software y necesarios para llevar adelante esta alternativa. Finalmente, el tiempo que demandaría el desarrollo de esta alternativa estaría dentro de los plazos determinados por la universidad para la realización del proyecto.

1.13.2.3. Factibilidad legal

Estimamos que esta alternativa es factible legalmente a pesar de que la mayoría de las herramientas utilizadas no son de uso libre, pero se cuenta con las licencias académicas brindadas por la Universidad. Que posteriormente deberán de ser compradas si el cliente opta por mantener el sistema.

1.13.2.4. Factibilidad Económica

Se determinaron los recursos para desarrollar, implantar, y mantener en operación el sistema programado, basado en evaluaciones donde se puso de manifiesto el equilibrio existente entre los costos intrínsecos de Azure y los beneficios que se derivaron de éste, lo cual permitió observar de una manera más precisa las bondades de implementar dicha solución.

Además, se cuenta con las herramientas necesarias para la puesta en marcha del sistema, solo requerirá de una pequeña inversión inicial que es insignificante comparado con una basada en servidor local.

1.13.3. Alternativa física 3

Como alternativa 3 se consideró una versión “en la nube” de bajo costo y *open source*. En el que se aprovecharía todas las ventajas del software libre pero la falta de compatibilidad entre los componentes podría generar algunos contratiempos.

Arquitectura: El sistema estaría basado en la arquitectura en la nube propuesta por Digital Ocean utilizando *frameworks* de uso libre como MonoProject y MySQL.

- **Ventajas:** La principal ventaja de implementar .NET en Mono Project es que esta tecnología ha sido liberada como un estándar de la ECMA. Todas las especificaciones están disponibles para cualquiera que quiera hacer una implementación. Con esto los desarrolladores de Mono se ahorraron una gran cantidad de esfuerzo en análisis y diseño, cosa que no pasaría si hubieran decidido crear una nueva plataforma de desarrollo. Además, trae la ventaja de aprovechar toda la campaña de promoción que Microsoft está gastando en esta tecnología para crear un software libre basado en una tecnología ampliamente documentada y de gran calidad.
- **Desventajas:** Existe el riesgo de que Microsoft reclame los derechos de C# utilizados por Mono Project. Por ahora no parece que Mono sea una gran amenaza para MS, es más, los desarrolladores del proyecto declaran que la gente de Redmond se ha mostrado como “buenos ciudadanos” frente al proyecto. Sin embargo, en el momento en que Mono se convierta en una amenaza, lo más probable es que MS pueda romper fácilmente la compatibilidad entre Mono y su implementación de .NET, vía patentes de software. De todas maneras, en el caso de que esto pase, lo cual es muy probable, no afectaría, de ninguna manera, el objetivo principal de Mono, lo único con lo que acabaría sería con algunos beneficios adicionales.

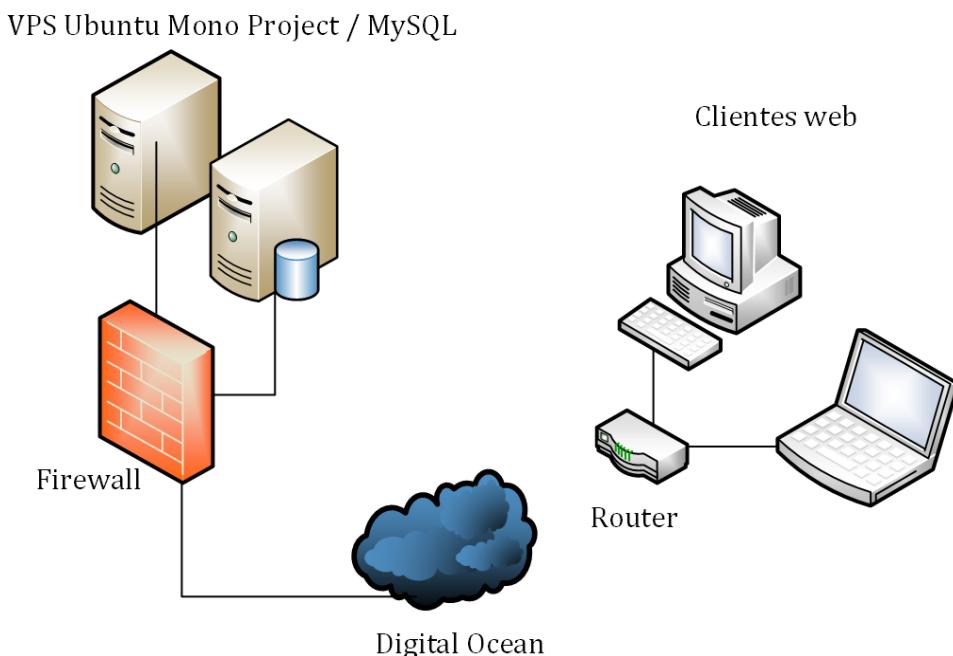


Fig. 29 Diagrama de alternativa física 3

<i>Equipo o producto</i>	<i>Cantidad</i>	<i>Especificación</i>
<i>Computadoras de testeo</i>	1	AMD FX-9590, RAM 4 Gb
<i>Backup</i>	1*	Amazon Glacier
<i>Sistema operativos</i>	1*	Cent OS con Mono Project 3.8
<i>Hosting</i>	1*	OceanDrive plan 40/mo
<i>IDE</i>	1*	Visual Studio 2013 Ultimate
<i>Conexión</i>	1	Internet fibra optica ANTEL
<i>DNS</i>	1	CloudFlare

*Licencias empresariales

Fig. 30 Tabla de alternativa física 3

1.13.3.1. Factibilidad operativa

Esta alternativa es factible operativamente ya que se une los beneficios que brindan tanto la tecnología de .Net y C# con el software libre propuesto Mono Project.

Se aprovecha el éxito de .Net y también las decenas de miles de libros, sitios web, tutoriales, y ejemplos de código fuente para ayudar en cualquier problema imaginable que pueda surgir a lo largo del proyecto.

Además, Mono Project tiene una amplia librería de clases que proporciona miles de clases incorporadas que podrán aumentar la productividad al desarrollar el sistema propuesto.

1.13.3.2. Factibilidad técnica

Se posee con algunas de los conocimientos necesarios de desarrollo, recursos de hardware y software necesario. Los cuales en este caso son brindados por la Universidad ORT.

Visto que esta es una plataforma relativamente nueva, se tendrá que planificar un tiempo de capacitación para la correcta utilización y despliegue que se estima que podría llegar a demorar 4 días.

1.13.3.3. Factibilidad legal

Se cuenta con las licencias legales dentro del marco académico para desarrollar la aplicación. Ya que esta es una solución libre esta alternativa no presenta ningún problema legal, siempre y cuando se utilice para lo descripto dentro de las licencias de derecho de autor.

1.13.3.4. Factibilidad Económica

Para este caso la alternativa es totalmente factible económicamente ya que no se necesita adquirir ninguna licencia, dado que el sistema será basado en una arquitectura *open source*.

Lo único a tener en cuenta es que se deberá adquirir el hosting OceanDrive el cual no presentaría ningún problema económico.

1.13.4. Opción técnica 1

Esta opción comprenderá las siguientes tecnologías:

- MVC
- Entity Framework

Para dar un pequeño vistazo a las tecnologías ASP.NET MVC se podría decir que es un nuevo *framework* para aplicaciones web creado por Microsoft, diseñado bajo la idea de la separación de responsabilidades y la posibilidad de implementar el *testing*. No posee ni *viewstate*, ni controles de servidor.

Esto se usará en conjunto con ADO.NET Entity Framework que proporciona varias ventajas a una aplicación. Una de las más importantes es la capacidad de utilizar el modelo conceptual para separar las estructuras de datos que usa la aplicación del esquema en el origen de datos. De esta forma se pueden realizar más adelante y con facilidad cambios en el modelo de almacenamiento o en el propio origen de datos sin realizar los cambios correspondientes en la aplicación.

Ventajas:

- **Arquitectura de proyecto:** Al estar fuertemente implementada la separación de responsabilidades, también tenemos una arquitectura del proyecto ordenada.
- **Desarrollo manejado por tests:** Los controles son clases separadas, por lo cual, es posible hacer test automáticos.
- **Reutilización:** Los controles no están atados a una vista, por lo cual pueden ser reutilizados.
- **Performance:** Las páginas son mucho más livianas comparadas con los *web forms*.
- **Control total del HTML:** Como no existen los controles de servidor, la única opción es utilizar los controles HTML, por lo que sabemos cómo se terminará renderizando la página. La integración con librerías JavaScript es realmente simple.
- **Soporte para trabajo en paralelo:** Al estar todo realmente separado, es posible que un desarrollador esté trabajando en una vista, mientras otro esté en el controlador y un tercero esté en el modelo, sin que interfieran entre ellos.
- **SEO, URL Routing y REST:** Las características de *routing* permiten utilizar interfaces REST.
- **Extensión:** Soporte para múltiples motores de vistas como aspx, razor, etc.
 - **Características preexistentes de ASP.NET:** Como está construido sobre el *framework* ASP.NET, provee características como autenticación, *caching*, *session*, etc

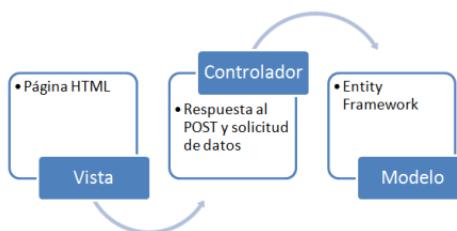


Fig. 31 Diagrama de patrón MVC

1.13.5. Opción técnica 2

Esta opción comprenderá las siguientes tecnologías:

- Web Forms
- ADO.NET

Por otro lado, los *web forms* aparecieron para solucionar los problemas que tenía la tecnología ASP clásica. Se creó un nuevo nivel de abstracción sobre la web sin estado, simulando el modelo con estado de los Windows *forms*. Se introdujeron conceptos como *postback* (un envío desde la misma página) y *viewstate* (un objeto que almacena los valores de los controles durante los *postbacks*) y se redujo la cantidad de código a escribir.

Ventajas:

- **Los Web Forms contienen controles de servidor ricos:** ASP.NET detecta el navegador del usuario y genera HTML y JavaScript apropiado para él. Además, controles como el *gridview* o el *listview*, contienen funcionalidad para enlazar datos.
- **Viewstate:** Normalmente los controles no retienen los valores entre los pedidos al servidor, pero en los *web forms* esto se logra guardando el último estado conocido en un campo oculto denominado *viewstate*.
- **Programación manejada por eventos:** Microsoft introdujo la programación manejada por eventos, con la cual, los desarrolladores no tenían que apoyarse sobre los métodos POST y GET para las interacciones con el servidor. Haciendo doble *click* sobre un control, se genera un bloque de código que manejará el evento de dicho control en el servidor; sin que el desarrollador tenga que saber qué es lo que ocurre.
- **Desarrollo de aplicaciones rápido:** Con los controles de servidor más el modelo manejado por eventos más *viewstate*, el desarrollador queda abstraído de muchas de las complejidades, con lo cual programa más rápido.
- **Curva de aprendizaje pequeña:** Es posible hacer aplicaciones con muy poco conocimiento de HTML y JavaScript.

Esto se usará en conjunto con ADO.NET clásico que permitirá elegir el diseño para la capa de persistencia. El cual es un conjunto de componentes del software que pueden ser usados por los programadores para acceder a datos y a servicios de datos. Es una parte de la biblioteca de clases base que están incluidas en el Microsoft .NET *framework*. Es comúnmente usado por los programadores para acceder y para modificar los datos almacenados en un sistema gestor de bases de datos relacionales, aunque también puede ser usado para acceder a datos en fuentes no relacionales.

1.13.6. Opción técnica 3

Esta opción comprenderá las siguientes tecnologías:

- GeneXus
- Work With Plus

Se evalúa como tercera opción técnica GeneXus, para describir básicamente ésta opción, podemos decir que es una herramienta de desarrollo de software ágil, multiplataforma, orientada principalmente a aplicaciones web empresariales, plataformas Windows y dispositivos móviles o inteligentes. Donde el desarrollador describe sus aplicaciones en alto nivel, de manera mayormente declarativa, a partir de lo cual GeneXus genera código para múltiples plataformas, las cuales incluyen las que se necesitarán para el proyecto del cliente.

Ventajas:

- **Interactivo:** en la construcción de aplicaciones.
- **Automático:** genera el 100% del código de aplicación.
- **Rapidez:** Para la creación de prototipos funcionales.
- **Reutilización de bases:** Aunque no es necesario para nuestro proyecto, GeneXus permite crear una nueva base de datos a partir de una ya existente.
- **Simplicidad:** Utiliza los recursos más avanzados de la inteligencia artificial, para que se utilicen de manera simple.
- **Desarrollo multi-usuario:** Si el equipo está trabajando en forma separada, permite que los miembros puedan trabajar de manera conjunta en el mismo proyecto.
- **Curva de aprendizaje nula:** Debido a que ya se cuentan con los conocimientos avanzados sobre esta tecnología y se trabaja diariamente con la misma.

Una buena razón por la cual se elijaría esta herramienta consiste en que las aplicaciones y sus bases de datos son cada vez más complejas, y al diseñar grandes bases de datos se cometen muchos errores humanos y, básicamente, porque en las grandes organizaciones no existe nadie que conozca los datos de la empresa con la adecuada objetividad y el suficiente detalle.

Además, la elección de ésta alternativa permite desarrollar aplicaciones que contengan altas, bajas y modificaciones, permitiendo en poco tiempo tener resultados a la vista. A su vez para realizar prototipos para un cliente antes de darle el producto final, lo cual no implica que lo ya generado no pueda ser usado como producto final. Por lo tanto, ayuda a la localización temprana de errores y un mejor seguimiento a los requerimientos de los usuarios, así se logra implantar aplicaciones en el menor tiempo posible y con la mayor calidad posible.

1.13.7. Selección de la alternativa

Dado los requerimientos expresados por el cliente y los conocimientos que se poseen se consideraron las tres alternativas físicas y técnicas propuestas anteriormente, se seleccionó la segunda alternativa física y la primer opción técnica.

Una de las principales características por la cual se optó por desarrollar el proyecto utilizando la primer opción técnica, que comprende utilizar el patrón MVC y Entity Framework se debe al aprendizaje personal que conlleva y al desafío que supone aprender una nueva tecnología que relativamente comparando con las demás opciones.

Se posee menos conocimientos y experiencia ya que en la segunda opción donde se utilizan *webforms* y ADO.NET se aprendió durante el transcurso de la carrera y la tercera se usa laboralmente.

Además, elegir esta opción técnica por sobre los *webforms* implica, que es la tecnología a donde se dirige la web por el momento, la mayoría de lenguajes tiene su *framework* MVC, Ruby tiene Rails, php tiene Cakephp, Code Igniter, Python tiene Djando y Pylon, Java tiene Springmvc y Grails. Si bien es cierto que ASP.NET *web forms* puede ahorrar tiempo, pero ese tiempo después a la larga se gastaría resolviendo otros inconvenientes que trae esa tecnología como el *viewstate* y la falta de testabilidad junto el desorden que permite a los desarrolladores, además el performance es mucho más bajo al de ASP.NET MVC.

Frente a la realidad de que es un proyecto académico y que es una opción con una complejidad alcanzable con lo dado dentro de la carrera se justicia la curva de aprendizaje a esta nueva tecnología.

Las soluciones de infraestructura como servicio a través de máquinas virtuales en la nube y otros componentes *cloud*, ofrecerán a la empresa una gran flexibilidad para mover los servidores (y aplicaciones) a la nube y aumentar o disminuir los recursos necesarios según sea necesario. Por esta razón y otras ventajas que ofrecen este tipo de servicios se decidió utilizar la segunda alternativa física.

De esta manera se aprovecha todas las características de integración que supone utilizar dicha infraestructura con el IDE que supone utilizar la opción técnica escogida. Este IDE (Visual Studio) posee varias funcionalidades que permiten trabajar de una manera flexible y ágil para crear y administrar servicios de Azure en muy poco tiempo y directamente desde la interfaz gráfica o consola de administración. Se puede iniciar sesión con una cuenta de Azure para ver al momento la información de los servicios e implementaciones actuales de proyectos, así como crear, administrar sitios web, servicios en la nube, iniciar, detener máquinas virtuales, actualizar y administrar datos, etc.

Enumerando las principales ventajas de esta solución son:

- **Reducción de costes:** A diferencia de invertir en adquirir y mantener el hardware de la empresa constructora, la opción nube te ofrece evitar la compra y un importante ahorro en consumo energético, soporte y mantenimiento. Por tanto, optando por esta alternativa, la empresa pasará a un modelo de costes basado únicamente en la capacidad que requieras en cada momento.
- **Flexibilidad:** La nube le permitirá obtener rápidamente más o menos recursos / potencia de servidor según la necesidad en cada momento, sin inversiones en activos.
- **Focalización:** La infraestructura como servicio le permitirá a la empresa constructora liberar tiempo y recursos hasta ese momento destinados a mantener su propia infraestructura, para dirigirlos a tareas de mayor valor para el negocio.
- **Urgencias, en minutos:** A diferencia de adquirir servidores o optar por otras soluciones en proveedores de internet clásicos, contar con un nuevo servidor en Windows Azure requiere solo de escasos minutos.
- **Disponibilidad:** Las soluciones de máquinas virtuales están basadas en una infraestructura de alta redundancia y la información está replicada en como mínimo 3 ubicaciones físicas. Todo ello permite asegurar la continuidad del servicio y disponibilidad de tus datos.
- **Seguridad:** Habitualmente, las medidas de seguridad más sofisticadas están fuera del alcance de la mayoría de empresas. No obstante, las normativas de seguridad que cumplen los *datacenters* de Microsoft proporcionarán al cliente la confianza de que sus datos estarán totalmente protegidos.

En resumen, esta selección se fundamenta en que cumple las necesidades explícitas del cliente y es factible desde todo punto de vista.

1.14. Análisis de riesgos

Como en todo proyecto existen riesgos en este se utilizará una estrategia proactiva para la administración del riesgo. Se comenzará mucho antes de iniciar el trabajo técnico. Los riesgos potenciales se identificarán, su probabilidad e impacto se valorarán y se clasificarán según importancia. Luego, el equipo de desarrollo descripto anteriormente establecerá un plan para gestionar el riesgo. El objetivo principal es evitarlo, pero, como no en todos los casos son evitables, este equipo trabajará para desarrollar un plan de contingencia que le permitirá responder en forma controlada y efectiva.

Se categorizará en dos tipos de riesgos: riesgos genéricos y riesgos específicos del producto. Según Pressman [2], los riesgos genéricos son una amenaza potencial a todo proyecto de software por otro lado los riesgos específicos del producto pueden identificarse solamente por quienes tienen clara comprensión de la tecnología, el personal y el entorno específico del software que se construye.

Los riesgos de este proyecto agrupados en subcategorías genéricas (sugeridas por Pressman) son:

- Tamaño del producto (TP):
 - Diseño poco realista
 - Implementación tediosa y más lenta de lo esperado
 - Mantenimiento poco eficiente
 - Especificaciones poco claras
 - Mayor número de usuarios de los previstos
 - Tráfico de datos alto
- Impacto empresarial (IE):
 - Presupuesto acotado
 - Normativas de la empresa constructora no tenidas en cuenta
 - Recursos limitados (hardware)
 - Cliente abandona
 - Mercado genera que el cliente pida más requisitos
 - Problemas legales
- Características de los participantes (CP):
 - Mala comunicación con el cliente
 - Los participantes finales se “resisten” al producto
 - Capacitación (documentación de ayuda) para participantes del cliente
- Definición del proceso (DP):
 - Cliente cambia los requisitos
 - Las fechas planeadas son muy ajustadas
 - La reutilización de código es pobre

- Entorno de desarrollo (ED):
 - Las herramientas son anticuadas
 - Falta de capacitación de personal
 - Licencias para herramientas caras
 - Problemas de compatibilidad
 - Problemas técnicos de terceros (*datacenter*, conexión u otros)
- Tecnología por construir (TC):
 - La tecnología utilizada no podrá alcanzar las expectativas deseadas
 - Se innovo demasiado o muy poco
 - Problemas de seguridad no previstos
- Tamaño y experiencia del personal (TEP):
 - Asignación y organización del personal poco realista
 - Conflictos entre el personal (medio ambiente pobre de desarrollo)
 - Perder la dirección debido al cambio de personal o de enfoque
 - Disminución del esfuerzo de personal

Al terminar el análisis de este los diferentes riesgos del proyecto, se puede afirmar que el equipo de desarrollo es capaz de realizarlo en el tiempo indicado según las estimaciones.

Dado que se dominan los conocimientos adquiridos en la carrera dictada por la Universidad, experiencias personales y que existen referencias de bibliografía para profundizar esos conocimientos.

En el caso de no dominar completamente las tecnologías, se cuenta con la disposición a investigar y consultar el material necesario o personas entendidas en el tema para la comprensión de ellas. Cabe destacar que se posee una relación laboral, lo que hace que disminuya su disponibilidad horaria para dedicar al proyecto, pero este aspecto se mitiga con la distribución horaria que se realizó para compensar esa diferencia.

1.14.1. Tabla de riesgos

Riesgo	Categoría	Probabilidad	Impacto	MMMR
Diseño poco realista	TP	50%	2	
Implementación tediosa y más lento de lo esperado	TP	60%	1	TP2
Mantenimiento poco eficiente	TP	10%	2	
Especificaciones poco claras	TP	20%	3	
Mayor número de usuarios de los previstos	TP	25%	3	
Tráfico de datos altos	TP	20%	4	
Presupuesto acotado	IE	60%	2	
Normativas de la empresa constructora no tenidas en cuenta	IE	30%	2	
Recursos limitados (hardware)	IE	10%	3	
Cliente abandona	IE	15%	1	IE4
Mercado genera que el cliente pida más requisitos	IE	30%	3	
Problemas legales	IE	40%	2	
Mala comulación con el cliente	CP	30%	4	
Los participantes finales se "resisten" al producto	CP	45%	4	
Capacitación (documentación de ayuda) para participantes del cliente	CP	30%	3	
Cliente cambia los requisitos	DP	70%	2	
Las fechas planeadas son muy ajustadas	DP	85%	2	DP2
La reutilización de código es pobre	DP	60%	2	
Las herramientas son anticuadas	ED	20%	3	
Falta de capacitación de personal	ED	15%	4	
Licencias para herramientas caras	ED	10%	4	
Problemas de compatibilidad	ED	30%	3	
Problemas técnicos de terceros (datacenter, conexión u otros)	ED	30%	2	
La tecnología utilizada no podrá alcanzar las expectativas deseadas	TC	10%	3	
Se innovo demasiado o muy poco	TC	30%	2	
Problemas de seguridad no previstos	TC	60%	2	
Asignación y organización del personal poco realista	TEP	50%	2	
Conflictos entre el personal (medio ambiente podre de desarrollo)	TEP	50%	2	
Perder la dirección debido al cambio de personal o de enfoque	TEP	30%	3	
Disminución del esfuerzo del personal	TEP	80%	2	

*Valores de impacto:

Fig. 32 Tabla de riesgos

1. Catastrófico
2. Crítico
3. Marginal
4. Despreciable

1.14.2. Plan de contingencia MMMR

A continuación, se describe un plan de mitigación, monitoreo y manejo de riesgo (MMMR) para aquellos riesgos que tengan una probabilidad alta o impacto crítico o catastrófico.

Los tres objetivos principales que se busca al realizar esto son; valorar si los riesgos predichos en efecto ocurren, asegurar que los pasos para evitar el riesgo definidos para un riesgo determinado se aplican de manera correcta y recopilar información que pueda usarse para futuros análisis de riesgos.

Hoja de información de riesgo

Riesgo ID: TP2	Fecha: 05/09/14	Probabilidad: 60%	Impacto:
Descripción: Puede ocurrir que al momento de implementar el proyecto también llamado desplegar, se presenten problemas u otros conflictos que enlentezcan esta etapa. Estos pueden repercutir en cambiar software, código o hardware. Esto se tiene que evitar ya que en este punto el proyecto está en su etapa final.			
Refinamiento/contexto: Subcondición 1 – Personal poco capacitado para esta tarea. Subcondición 2 – No se efectuó la etapa de experimentación correctamente. Subcondición 3 – Debido a que es una de las últimas etapas el equipo de despliegue se enfrentara a la falta de tiempo.			
Mitigación/monitoreo: 1. Contactar al equipo de despliegue que realizo la etapa experimental. 2. Verificar que el hardware sea el especificado en los requerimientos. 3. Verificar si el problema ocurre con el proyecto en estado experimental (que está funcionando). 4. Realizar la menor cantidad de cambios posibles.			
Manejo/plan de contingencia/disparador: <ul style="list-style-type: none">• ER calculada en 800U\$S, asignar esta cantidad dentro de los costos de contingencia del proyecto.• Realizar revisiones contactando al equipo de despliegue experimental y realizar la mínima cantidad de cambios posibles para asegurar el funcionamiento normal.• Disparador: <<Todavía no disparado>> XX/XX/XX.			
Estado actual: Inactivo.			
Originador: Gabriel Martínez		Asignado: Equipo de despliegue	

Fig. 33 Hoja de información de riesgo TP2

Hoja de información de riesgo

Riesgo ID: IE4 **Fecha:** 05/09/14 **Probabilidad:** 15% **Impacto:** Catastrófico

Descripción: En este proyecto al igual que en todos, la posibilidad de que el cliente abandone es un factor que puede generar pérdidas incalculables. Para ello se tienen que tomar medidas preventivas. Mantener al cliente conforme y asegurar una comunicación activa.

Refinamiento/contexto:

Subcondición 1 – Cliente insatisfecho.
Subcondición 2 – Problemas económicos.
Subcondición 3 – Competencia.

Mitigación/monitoreo:

1. Se contacta el ingeniero/gerente de proyecto.
2. Se contacta al cliente o se renegocia.
3. Cesar actividad según etapa.
4. Intentar sacar provecho de los módulos funcionales que pueden ser vendidos a un tercero.
5. Vender recursos físicos utilizados.

Manejo/plan de contingencia/disparador:

- ER calculada en 150000\$S, asignar esta cantidad dentro de los costos de contingencia del proyecto.
- El gerente de proyecto renegocia o en su defecto se maximiza la reutilización de código para futuros proyectos.
- Disparador: <<Todavía no disparado>> XX/XX/XX.

Estado actual: Inactivo.

Originador: Gabriel Martínez

Asignado: Equipo de despliegue

Fig. 34 Hoja de información de riesgo IE4

Hoja de información de riesgo

Riesgo ID: DP2 **Fecha:** 05/09/14 **Probabilidad:** 85% **Impacto:** Crítico

Descripción: El planeamiento del proyecto no es algo para hacerse solamente una vez al comienzo del proyecto. Se debe observar el progreso de cada equipo o personal y actualizar adecuadamente el plan de proyecto debe ser una tarea constante del gerente del proyecto (ingeniero a cargo).

Refinamiento/contexto:

Subcondición 1 – Mala planificación del proyecto.
Subcondición 2 – Estimaciones erróneas.
Subcondición 3 – Personal no cumplió las expectativas del gerente

Mitigación/monitoreo:

1. Determinar las condiciones exactas para que el proyecto sea terminado.
2. Revisar el inventario de todo el trabajo que se debe hacer.
3. Decidir si el plan tiene sentido o concuerda con la realidad.
4. Revisar dependencias entre tareas, lo que lleva a esta situación.
5. Asignar personal extra especializado en la tarea.

Manejo/plan de contingencia/disparador:

- ER calculada en 4000\$S, asignar esta cantidad dentro de los costos de contingencia del proyecto.
- El gerente de proyecto ajusta las fechas o en su defecto se contrata nuevo personal solo si este realmente acelera la productividad (ej: especialista con experiencia en la tarea atrasada)
- Disparador: <<Todavía no disparado>> XX/XX/XX.

Estado actual: Inactivo.

Originador: Gabriel Martínez

Asignado: Equipo de despliegue

Fig. 35 Hoja de información de riesgo DP2

1.15. Plan de Proyecto

1.15.1. Definición del Proceso

1.15.1.1. Metodología

Como se ha mencionado anteriormente, para el proyecto se decidió utilizar una adaptación de la metodología ágil de desarrollo denominada *scrum*, la cual se adapta perfectamente a las necesidades y desafíos que el proyecto comprende, utilizando el tiempo disponible de la manera más provechosa.

Para ello está metodología establece distintas pautas para los aspectos del desarrollo del Proyecto, como los roles y responsabilidades, organización, mediciones, utilización de herramientas, entre otros. Esta metodología tiene como característica que la planificación se divide en iteraciones de desarrollo llamadas *sprints* que se desarrollan por un período de tiempo de una quincena. Estos *sprints* contienen distintas tareas a realizar y los chequeos necesarios para determinar si se cumplieron correctamente. Como beneficios proporcionados por la utilización de esta metodología se destacan: la gestión regular de las expectativas del cliente basado en aspectos tangibles, se obtienen los resultados anticipados al finalizar cada sprint (prototipos), flexibilidad y adaptación respecto a las necesidades del cliente o cambios en el mercado, gestión sistemática del retorno de inversión, mitigación sistemática de los riesgos del proyecto, entre otros. En la documentación, se detallan las distintas tareas a realizar en los ocho *sprints* previstos para la realización del proyecto.

1.15.1.2. Ciclo de vida elegido

Ya que existen distintos modelos de ciclo de vida y que no existe uno que funcione ante cualquier proceso, para su elección deben estudiarse las características del proyecto a realizar. Dentro de las características a estudiar se encuentra la naturaleza y disponibilidad del cliente, la disponibilidad para correr riesgos, el dominio de la aplicación, los tiempos de trabajo, la volatilidad el grado de comprensión de los requerimientos del cliente. Luego del análisis de estas características se llegó a la elección de un modelo de ciclo de vida incremental e iterativa, en el que se agrupan las tareas a realizar en un número determinado de iteraciones. Entre las ventajas fundamentales de su utilización se encuentran la obtención de productos operativos intermedios que pueden ser presentados al cliente a modo de informe de progreso, la flexibilidad de adaptación al cambio de planificación y disminución de impacto de los cambios de requerimientos. Cada etapa comienza con la visión general del producto, se especifican y detallan las funcionalidades ordenadas por prioridad de mayor a menor que puedan realizarse en dicha iteración. Cada una finaliza con la entrega de una parte operativa del producto al cliente.

1.15.2. Incrementos o iteraciones definidas

Los incrementos o iteraciones generalmente están descriptas en la lista de tareas como las entregas de los *sprints*, que suponen un producto listo comprobable por el cliente, gracias a esto se podrá revisar el avance con el cliente y mantener un orden donde se espera que cada resultado de sprint esté completo para ser comunicado con el cliente entregando un informe, documentación o parte del sistema según corresponda.

1. El primer avance al cliente será cuando la identificación de necesidades y beneficios se complete el 20/03/15, en esta instancia se le entregará un resumen con las principales características del proyecto; necesidades principales, recursos a utilizar, beneficios esperados y el enunciado del producto.
Esta entrega es como un enunciado general del trabajo que se realizará y no implica detalles técnicos solo lo necesario para que el cliente monitoree el desarrollo del proyecto.
2. El segundo avance será el 07/05/15 cuando el análisis de requerimientos de software se complete, lo que significaría que ya se produjo, el análisis de requerimientos, una estimación de presupuestos, cronograma de trabajo entre otros que podrían ser comunicados al cliente en una reunión opcional liderada por el equipo de relaciones. Esta entrega corresponde con el fin del sprint 2 el cual será entregado al cliente para su posterior verificación.
3. En la etapa de diseño se necesitará una activa participación del cliente para la revisión de especificaciones, devoluciones y corrección, al finalizar el sprint 4 el 04/06/15 se presentará un informe con las principales características de esta etapa y el producto generado hasta el momento como resultado de este sprint.
4. En la etapa que se finaliza el sprint 6 se realizará otra entrega correspondiente al día 16/07/15 con la documentación de ayuda, esto le servirá al cliente a conocer el funcionamiento del sistema y que capacitación deben tener los posibles usuarios para utilizarlo, así como el producto correspondiente al fin de este sprint.
5. Esta entrega es una de las más importantes y consistirá en la pre visualización en estado experimental del proyecto por el cliente, esta entrega será liderada por el equipo de despliegue y está planeada para el 27/08/15 en un “laboratorio” que simulará las condiciones de operación del proyecto final.
6. La entrega final se estima para el día 21/10/15 en la etapa denominada “reunión final con el cliente” y se establecerán detalles finales, plan de mantenimiento y posibles capacitaciones a usuarios del sistema.

1.15.3. Integrantes y roles

Según Roger S. Pressman [3], el “factor humano” es un pilar muy importante para la administración efectiva de un proyecto de software. Y está comprendida dentro de las cuatro P: personal, producto, proceso y proyecto.

Es muy importante la elección adecuada del personal ya que el gerente que olvida que el trabajo de la ingeniera del software es una empresa intensamente humana nunca triunfará en la administración del proyecto. Aquel que se embarque sin un plan sólido pone en peligro el éxito del proyecto.

El Software Engineering Institute reconoce que “toda organización requiere mejorar continuamente su habilidad para atraer, desarrollar, motivar, organizar y conservar la fuerza de trabajo necesaria a fin de lograr sus objetivos empresariales estratégicos”.

A continuación, se describirá los roles necesarios para este proyecto y los perfiles que se necesitarán para organizar el factor humano de forma objetiva:

- ❖ Es importante aclarar que como este es un proyecto académico la implementación de dichos roles serán realizadas por los integrantes del proyecto, en una instancia más avanzada de desarrollo del sistema, posterior a la entrega académica se podrían necesitar más integrantes para cumplir con los roles especificados.
- Equipo de relaciones: este equipo tiene que estar formado con profesionales que aporten habilidades de relación cliente-empresa que sirva como un mediador y entienda las necesidades y especificaciones que el cliente proponga. Como se puede observar en el cronograma de trabajo algunas de sus tareas asignadas serán, por ejemplo:
 - Reuniones con cliente
 - Identificación de necesidades principales
 - Obtener aprobaciones del cliente
 - Comunicar a los documentadores las especificaciones del cliente

Es importante destacar que este equipo de relaciones es vital para que el proyecto esté bien encaminado, que sirva como un punto de entrada al cliente y de salida al entregar o presentar posibles iteraciones definidas o “partes del sistema” para mantener una relación cliente-empresa provechosa para ambos.

Este equipo puede estar formado por integrantes que ejerzan otros roles y que tengan un buen nivel de comunicación y experiencia en relaciones con clientes.

- Ingeniero/s en sistemas: Serán quienes deben planificar, motivar, organizar y controlar a los demás profesionales. Su tarea principal es la gerencia general del proyecto.

De vez en cuando deberán ejercer el papel de gerentes ejecutivos definiendo los temas empresariales de gran influencia sobre el proyecto.

Algunas tareas serán:

- Definir recursos a utilizar
- Estimación de presupuestos
- Crear un cronograma de trabajo
- Obtener aprobaciones del cliente (guiando al equipo de relaciones)

Se destacará su rol como líderes de equipo que implica mucho trato con la gente y una mezcla justa de habilidades personales. Según el modelo de liderazgo de Jerry Weingber este individuo debe contar con características de motivación que le permitan “alentar mediante un “empuje o jalón” al personal técnico a producir en su máxima capacidad.

A parte tener una buena habilidad para organizar, moldeando los procesos existentes o inventar nuevos que permitirán que el concepto inicial se traduzca en un producto final, siendo este el sistema de la constructora IGAFAL S.A terminado.

Aportar ideas o innovación alentando a las personas a crear y sentirse creativas, resolver problemas, diagnosticar conflictos técnicos y organizativos. Esto no se refiere a “solucionarlos” personalmente sino a estructurar sistemáticamente una solución o motivar adecuadamente a otros profesionales para desarrollarla, aplicando su experiencia en este tipo de proyectos si se presentan problemas adversos.

Por último deberá tener identidad administrativa, generar un sistema de logros para recompensar la iniciativa, optimizar la productividad de un equipo.

- Equipo de software: Estará dividido en analistas, desarrolladores, testeadores, equipo de despliegue y documentadores. La idea es usar el “paradigma síncrono” que es uno de los paradigmas que sugiere Constantine [Con93] para apoyar la compartimentalización natural de un problema y organizar a los miembros del equipo para trabajar.
Una de las citas que describirían perfectamente lo que se quiere lograr en cuanto al enfoque de este equipo, según un autor desconocido es “Si quieras ser incrementalmente mejor: sé competitivo. Si quieras ser exponencialmente mejor: sé cooperativo.” Por lo tanto, se busca que los miembros del equipo deben tenerse confianza entre sí, para garantizar esto se dividió el equipo de software según las habilidades para que las distribuciones de las mismas sean adecuadas al problema.

Los analistas tendrán tareas como:

- Investigar tareas a implementar
- Realizar la especificación de requerimientos
- Incorporar las especificaciones al proyecto
- Desarrollar especificaciones funcionales

Los desarrolladores podrán ser conformados también por analistas que en el momento no tengan ninguna asignación en las tareas mencionadas anteriormente realizarán:

- Primero que nada, leer y entender las especificaciones de los analistas
- Desarrollar el código según lo entendido
- Dividirse las subtareas entre ellos para desarrollar más cohesivamente

Los testeadores realizarán sus tareas cuando las anteriores asignaciones estén cumplidas por lo tanto este equipo podría también estar formado por desarrolladores, analistas y profesionales que se especialicen en *testing*. Realizaran lo siguiente:

- Primer *debugueo* general de código
- Identificar anomalías y modificar código
- Revisar integración de módulos

Los documentadores realizaran sus tareas en paralelo a los demás miembros y a lo largo de la duración del equipo de software, llevaran un seguimiento y documentaran todo lo que sucede en el proyecto y lo que se deberá realizar. Algunas tareas son:

- Desarrollar un SRS
 - Desarrollar documentación de ayuda
 - Revisar documentación con el cliente si corresponde
 - Incorporar correcciones o comentarios
- Equipo de despliegue, como el nombre sugiere estos realizaran sus tareas al final del proyecto en dos modalidades, de forma “experimental” y de forma “final”. Por ejemplo, tendrán que realizar:
 - Instalar software de forma experimental
 - Obtener comentarios del equipo
 - Evaluar posibles cambios a futuro
 - Determinar estrategia de despliegue final
 - Asegurar recursos para el despliegue final
 - Desplegar software

1.15.4. Plan de SQA

El equipo de SQA (aseguramiento de calidad) participa en la revisión de los productos desarrollados por los desarrolladores para determinar si son conformes o no a los procedimientos, normas o criterios especificados. Como se puede observar en el cronograma de trabajo en la sección

“Aseguramiento de calidad” el primer ítem denominado “fijar revisiones y auditorías” es en donde se determinará qué actividades debe realizar el grupo de aseguramiento de calidad a lo largo de todo el proceso de desarrollo de software, para asegurar que se están teniendo en cuenta las especificaciones de calidad especificadas para el proyecto.

A lo largo de todo el proceso de desarrollo y por cada vez que se asigna un subdesarrollo como completo, por ejemplo “Desarrollar dominio” -> “Dominio completo” esto generara que el equipo de desarrollo realice un plan de SQA y posteriormente lo efectué. Esto se cumple para cada *subítem* que este identificado como “Desarrollo... completo”.

1.15.5. Estándares definidos y convenciones

Continuación se describirá todos los estándares a utilizar.

- Para la documentación SRS se usará el estándar IEEE 830
 - 830-1998 — IEEE Recommended Practice for Software Requirements Specifications. 1998. [doi :10.1109/IEEESTD.1998.88286](https://doi.org/10.1109/IEEESTD.1998.88286)
- Para el código se usará nomenclatura sugerida por Microsoft para C#
 - <http://msdn.microsoft.com/library/ff926074.aspx>
- Para los planes de contingencia MMMR (RMMM por sus siglas en inglés)
 - <http://www.mhhe.com/engcs/compsci/pressman/graphics/Pressman5sepa/common/cs2/rmmm.pdf>
 - <http://www.rspa.com/docs/riskmmm.html> - Pressman

1.15.6. Plan de testing

La meta es encontrar errores con el mínimo esfuerzo posible, por lo tanto, solo se codificarán pruebas que tengan una alta probabilidad de encontrar uno. Estas pruebas en sí mismas deben de intentar abarcar un conjunto de características que logren la meta de encontrar la mayor cantidad de errores con el mínimo esfuerzo y tiempo.

Se estima que una fracción de los testeadores se dedique a pruebas de caja negra, las cuales son aquellas que se llevan a cabo en la interfaz del software, examinando algunos aspectos fundamentales del sistema con poca preocupación por la estructura lógica interna del software. Y otra fracción se dedique a pruebas de caja blanca donde se basarán en el examen cercano de los detalles del código. Las rutas lógicas a través del software y las colaboraciones entre componentes.

Al realizar las pruebas se deberá de tener en cuenta los siguientes puntos:

1. Realizar pruebas de caja blanca para tener una idea general de lo que se quiere probar.
2. Planificar pruebas de caja negra analizando valores límite.
3. Identificar clases de entrada y salida añadiendo más pruebas de caja negra para contemplar valores normales (los casos que generalmente ocurren).
4. Añadir pruebas basadas en valores límite. A partir de la experiencia y el sentido común de nuestro equipo de SQA, se adentrarán en situaciones que parecen proclives a padecer defectos, y se buscan errores en esos puntos.
5. Medir la cobertura de que se ha logrado con las fases previas y añadir más pruebas de caja blanca o negra si es necesario.

En el siguiente fragmento del cronograma de Gantt se puede observar cómo se deberá proceder con la fase de puramente de testeo nótese también que posiblemente el equipo de testeadores compartan integrantes con el de desarrolladores por lo tanto se dejó espacio entre medio para el desarrollo:

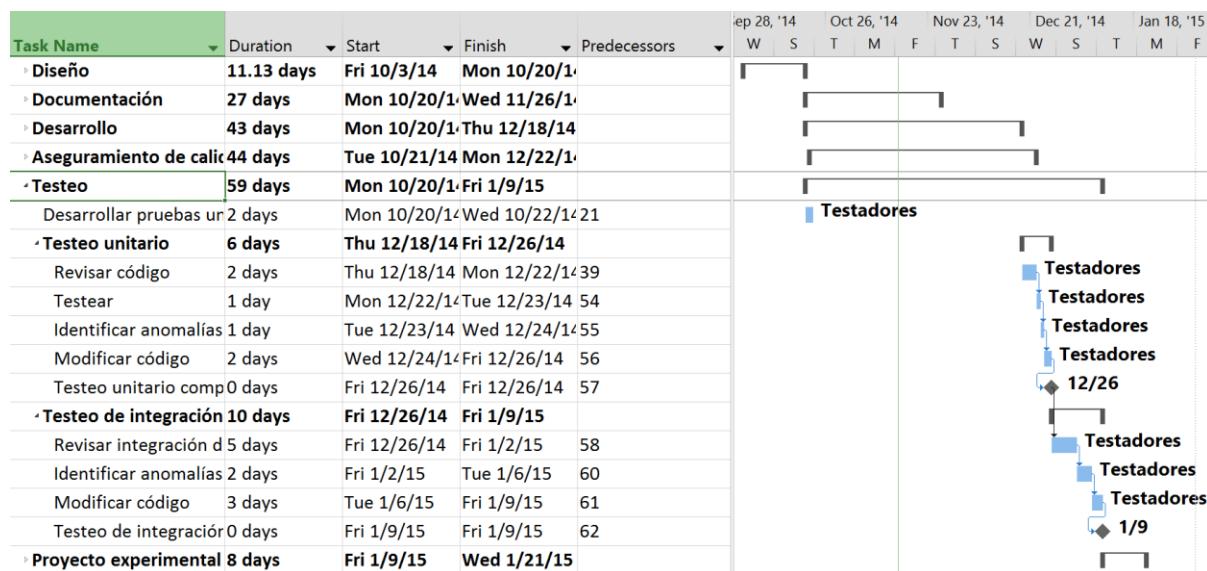


Fig. 36 Diagrama de Gantt (2014) para el testing

Cuando se implementó la metodología de *scrum* el diagrama resultante fue:

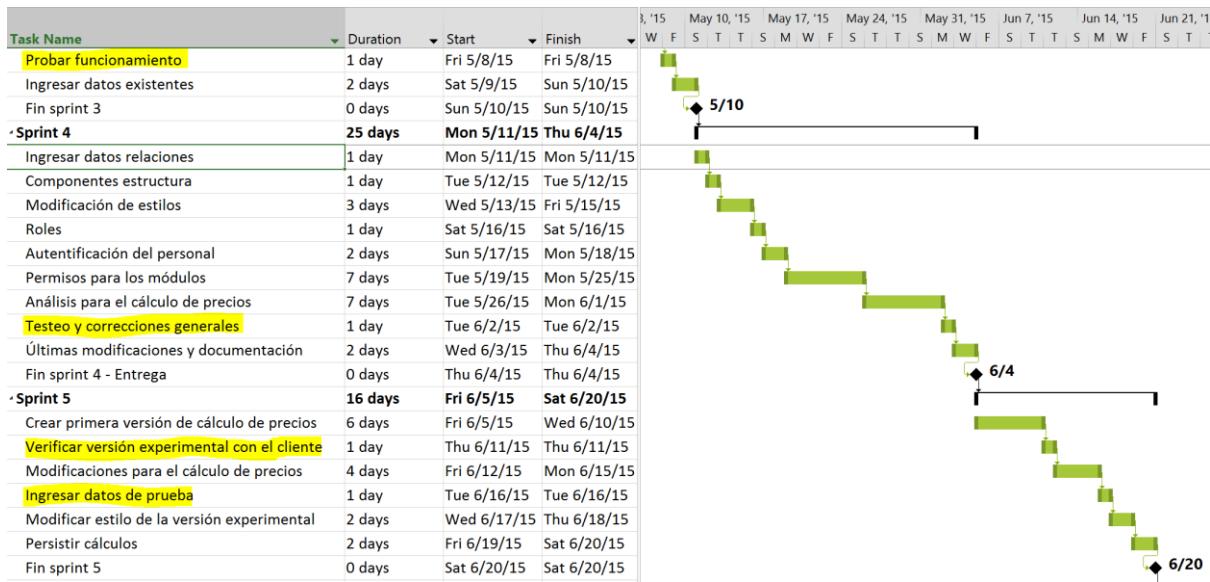


Fig. 37 Diagrama de Gantt para el testing

Es interesante destacar que el cambio de metodología de desarrollo impactó en la fase de testeo, donde previamente con la metodología clásica se tenía una larga etapa simultánea al desarrollo y en el *scrum* el testeo se ubicó inmediatamente después de desarrollar, antes de terminar con cada producto entregable del sprint. Además, se tuvo que diagramar teniendo en cuenta que ya no será un proyecto guiado por varios equipos sino un proyecto del tipo académico de máximo dos personas.

Aclaramos por este medio que no es posible garantizar que este sistema jamás falle, tan solo podremos realizar pruebas que disminuyan este riesgo.

1.15.7. Plan de SCM

El control del cambio es vital. Pero las fuerzas que lo hacen necesario también lo hacen desconcertante. Nos preocuparemos por el cambio porque una pequeña perturbación en el código puede crear una gran falla en el producto. Pero también puede corregir un gran fallo o permitir maravillosas nuevas capacidades.

Se tendrán varios *backups* dependiendo de la configuración de herramientas que se escoja (definidos en el punto 1.13).

Si se elige la versión “independiente” la herramienta de *backup* será Symantec Backup Exec la cual realizará *backups* periódicos cada 2 días donde los mismos serán alojados en un servidor de gran capacidad, *hosteado* localmente o usando unos más confiables en Amazon S3.

Si en cambio se elige la versión “en la nube” se utilizará la herramienta de *backups* automáticos basados en *droplets* de OceanDrive. Estos tendrán un costo sensiblemente inferior al cliente, pero como los mismos están basados en una arquitectura cerrada lo que significa que los mismos solo funcionaran en los servidores de OceanDrive, se recomienda también utilizar Symantec Backup Exec para tener una redundancia que imposibilite la perdida de software, ya sea por parte del hosting o por error del personal. Dicho en otras palabras, la metodología de *backup* será la misma que la versión “independiente” pero brindando un extra de seguridad que brinda la plataforma de OceanDrive con su arquitectura basada en *droplets*.

Dentro de la versión “en la nube” se puede optar por Azure, siendo ese el caso el procedimiento de *backup* será de forma análoga a lo mencionado anteriormente.

1.15.8. Plan de capacitación

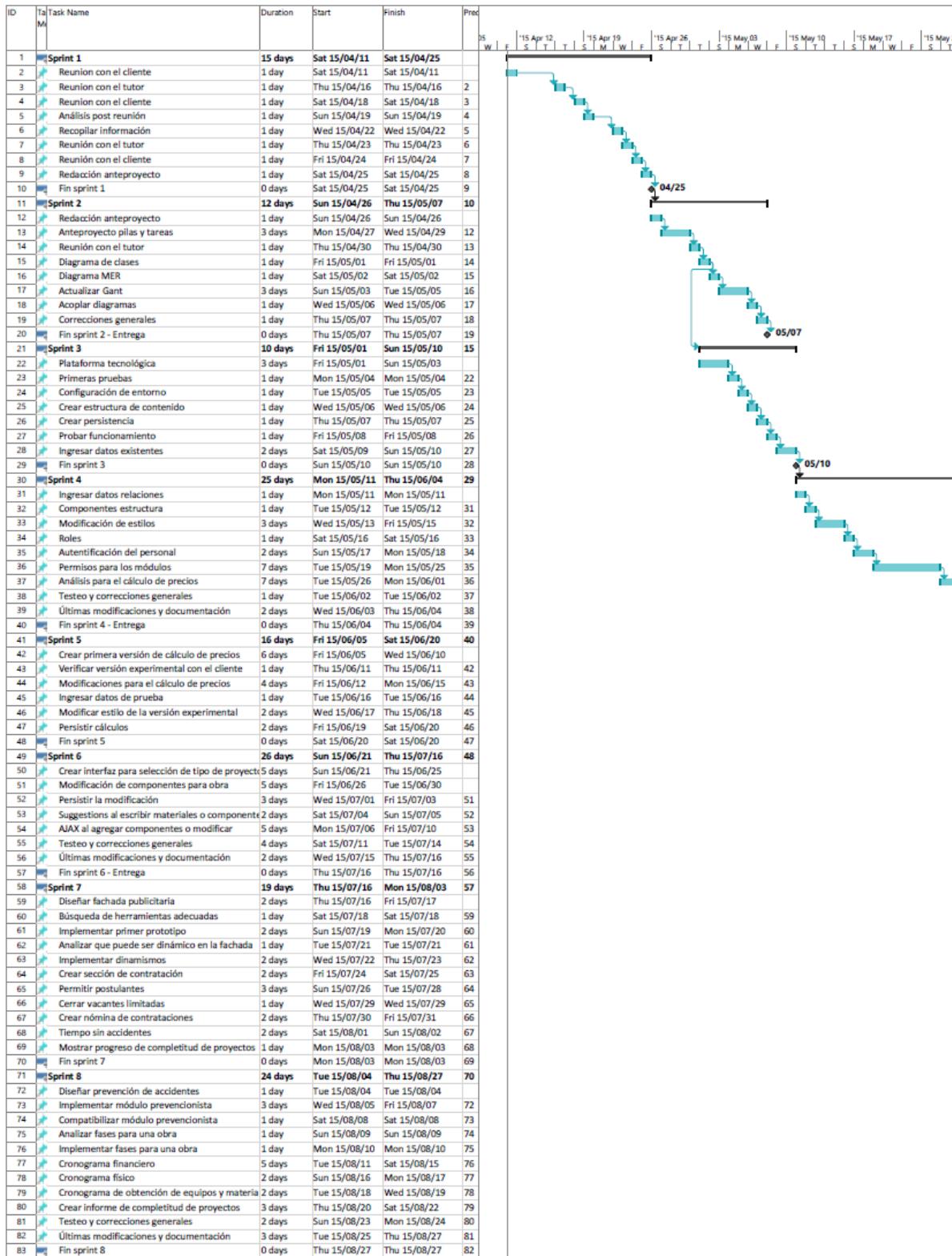
El día 21/10/15 según el cronograma de trabajo, se planificó dentro de lo que es la reunión final con el cliente, un plan de capacitación para los usuarios finales del sistema, este consiste dependiendo de si la propuesta se acepta o no por la empresa constructora en la apertura de un curso liderado por el equipo de relaciones en la utilización del sistema.

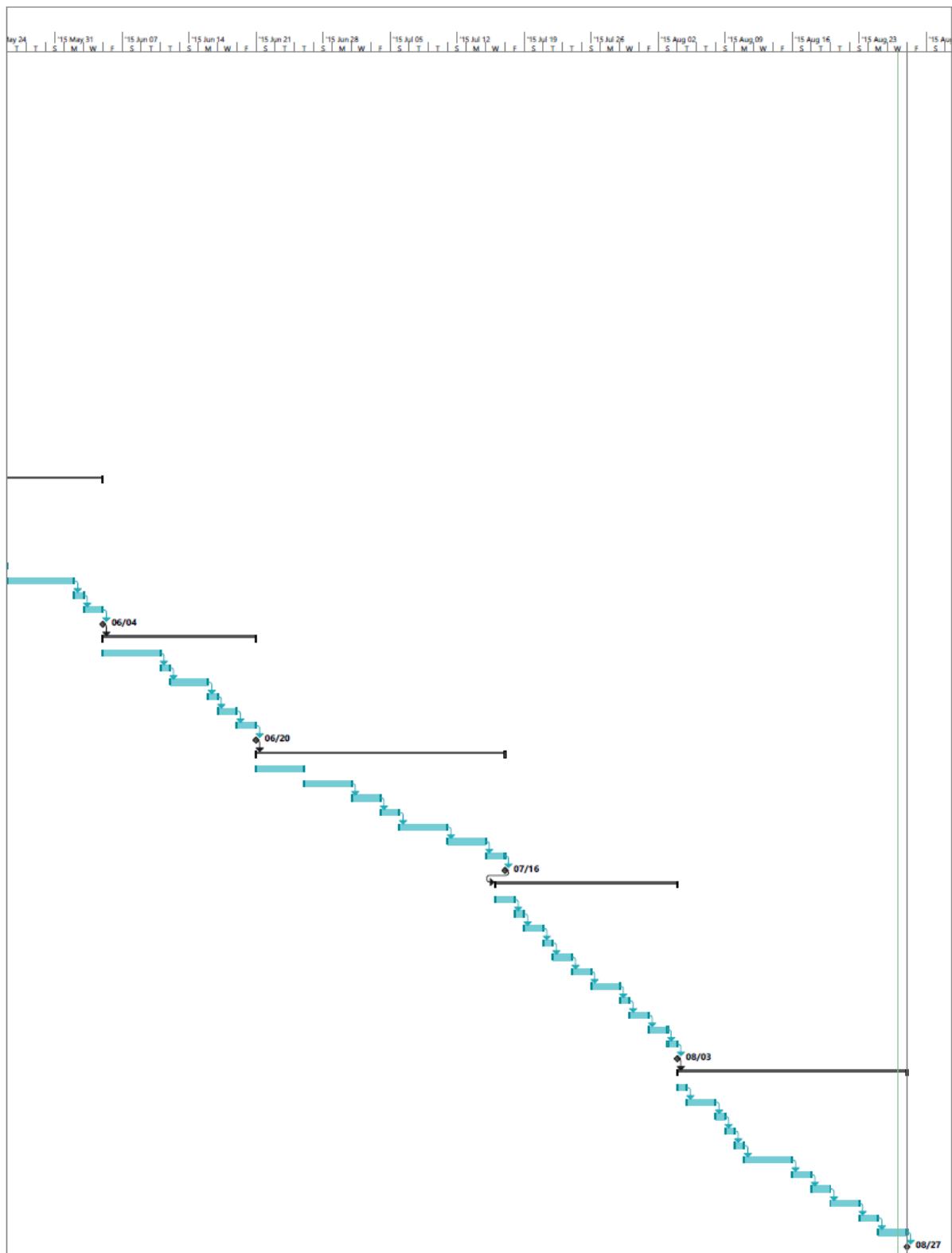
Este curso de capacitación será opcional y tendrá la duración que sea necesaria, dependiendo de los conocimientos de los usuarios y la cantidad de usuarios finales.

1.15.9. Cronograma de trabajo y criticidad

Se utilizó un cronograma de Gantt, en él se muestra las etapas de las que se compone el sistema para la constructora IGAFAL S.A diseñado y las actividades planificadas para desarrollarlo. Para ello, se señaló el principio y final de cada una de estas fases por medio de diagramas de barras. De esta forma, se puede visualizar fácilmente el calendario global que tendrá el proyecto.

Se recomienda ver anexo Cronograma de trabajo.ppt





1.15.9.1. Tiempos estimados según cronograma de Gantt

A continuación, se anexa un resumen de los tiempos estimados para cada sprint del proyecto.

Task Name	Duration	Start	Finish	Predecessors
▷ Sprint 1	15 days	Sat 15/04/11	Sat 15/04/25	
▷ Sprint 2	12 days	Sun 15/04/26	Thu 15/05/07	10
▷ Sprint 3	10 days	Fri 15/05/01	Sun 15/05/10	15
▷ Sprint 4	25 days	Mon 15/05/11	Thu 15/06/04	29
▷ Sprint 5	16 days	Fri 15/06/05	Sat 15/06/20	40
▷ Sprint 6	26 days	Sun 15/06/21	Thu 15/07/16	48
▷ Sprint 7	19 days	Thu 15/07/16	Mon 15/08/03	57
▷ Sprint 8	24 days	Tue 15/08/04	Thu 15/08/27	70

Fig. 38 Diagrama de tiempo de sprints

1.15.10. Plan de instalación

Para una correcta instalación del sistema es vital que la etapa de instalación denominada “proyecto experimental” sea realizada exitosamente por el equipo de despliegue y supervisada por el ingeniero a cargo, la cual debe permitir detectar o no posibles problemas antes de la instalación final, arreglarlos y obtener comentarios del equipo de desarrollo y del cliente.

Evaluados e implementados los posibles cambios al sistema se procederá a la etapa de despliegue.

En esta etapa lo primero que se realizará será una estrategia de despliegue que consiste en; asegurar los recursos necesarios, asignar tareas al personal de despliegue y desplegar el software según corresponda.

1.15.11. Diagrama de despliegue

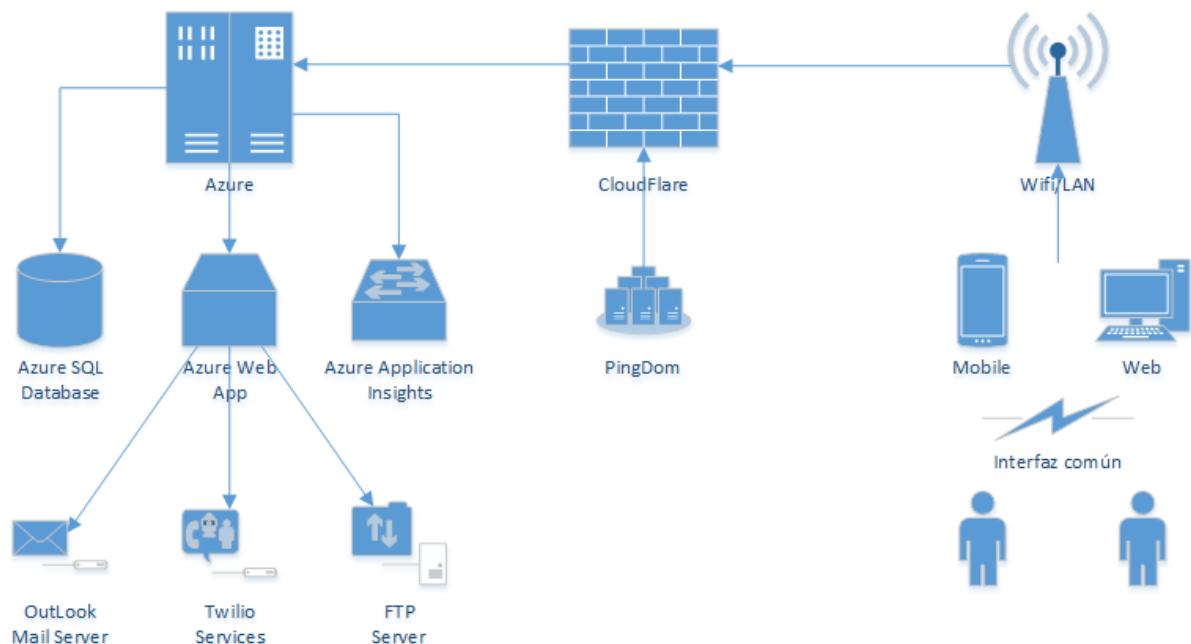


Fig. 39 Diagrama de despliegue

2. Desarrollo

2.1. Introducción

En el siguiente capítulo se describirá las etapas del desarrollo del software en orden cronológico. La descripción muestra el énfasis en la implementación de la metodología de gestión *scrum*. Se mostrarán las especificaciones de requerimientos de sistema, así como las tareas realizadas en los diferentes *sprints*.

Es de destacar que el resultado del *sprint 2* fue el documento anteproyecto, no existió un plan de proyecto en el correr del *sprint 1* y *2* por lo que se pasará a describir el desarrollo desde el *sprint 3*.

2.2. Sprint 1

2.2.1. Reuniones

2.2.1.1. Relevamiento inicial

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras
<i>Lugar</i>	Oficina de la empresa IGAFAL S.A
<i>Fecha</i>	2015/04/11 (2 Horas)
<i>Número de iteración / sprint</i>	1
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Arq. Jorge Carámbula Igarabide
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Arq. Jorge Carámbula Igarabide

<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none">• Introducción al proyecto• Relevamiento inicial	<ul style="list-style-type: none">• Algunos conocimientos generales sobre proyecto arquitectónicos	<ul style="list-style-type: none">• Leer sobre el tema

Fig. 40 Tablas de reunión de relevamiento inicial

2.2.1.2. Presentación del proyecto y análisis

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/04/16 (2 Horas)	
<i>Número de iteración / sprint</i>	1	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Presentación del proyecto al tutor • Análisis inicial 	<ul style="list-style-type: none"> • Se dificultó un poco la presentación de la idea principal del proyecto al tutor, debido que el anteproyecto estaba en etapas tempranas • Algunas veces se mencionó cuestiones de programación no relevantes para la reunión • Se buscó a muy grandes rasgos y no lo primordial • Ocurrieron algunas distracciones con la búsqueda de algunos requerimientos que al final no eran del todo relevantes 	<ul style="list-style-type: none"> • Continuar con la redacción del anteproyecto focalizando y buscando una idea principal • Intensificar y relevar más incisivamente la próxima reunión con el cliente.

Fig. 41 Tablas de reunión de presentación de proyecto

2.2.1.3. Relevamiento de las necesidades del cliente más incisivamente

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Oficina de la empresa IGAFAL S.A	
<i>Fecha</i>	2015/04/18 (3 Horas)	
<i>Número de iteración / sprint</i>	1	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Relevamiento de las necesidades del cliente • Esquemas conceptuales 	<ul style="list-style-type: none"> • Algunas observaciones del tutor fueron olvidadas durante el relevamiento 	<ul style="list-style-type: none"> • Anotar y resumir preguntas para la próxima reunión • Generalizar preguntas

Fig. 42 Tablas de reunión de relevamiento de las necesidades del cliente más incisivamente

2.2.1.4. Visita a la intendencia y cámara de la construcción por información

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Intendencia y cámara de la construcción	
<i>Fecha</i>	2015/04/22 (5 Horas)	
<i>Número de iteración / sprint</i>	1	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se consiguió información sobre listas de precios oficiales, rubros y un presupuesto de SASEM de ejemplo 	<ul style="list-style-type: none"> • Tiempo de espera 	<ul style="list-style-type: none"> • Analizar el material obtenido

Fig. 43 Tablas de reunión de visita a la intendencia y cámara de la construcción por información

2.2.1.5. Análisis del proyecto en base a la información conseguida

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/04/23 (2 Horas)	
<i>Número de iteración / sprint</i>	1	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se encontró una problemática del cliente a resolver • Se presentó la información obtenida al tutor y se pudo focalizar mejor el proyecto 	<ul style="list-style-type: none"> • Faltó esquematizar los problemas 	<ul style="list-style-type: none"> • Realizar esquemas y listas de tareas

Fig. 44 Tablas de reunión de análisis del proyecto en base a la información conseguida

2.2.2. Reunión retrospectiva

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Via skype	
<i>Fecha</i>	2015/04/24 (1 Hora)	
<i>Número de iteración / sprint</i>	1	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se pudo coincidir con las ideas académicas del proyecto y del tutor con las necesidades del cliente 	<ul style="list-style-type: none"> • Sigue faltando esquematizar los problemas 	<ul style="list-style-type: none"> • Realizar esquemas y listas de tareas

Fig. 45 Tablas de reunión de retrospectiva

2.3. Sprint 2

2.3.1. Análisis del anteproyecto

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/04/30 (2 Horas)	
<i>Número de iteración / sprint</i>	2	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none">• Se pudo obtener un ante proyecto inicial que servirá como base para desarrollar el proyecto	<ul style="list-style-type: none">• Algunos requerimientos no fueron especificados, si bien muchos se consideran obvios, se necesitará especificarlos en el anteproyecto	<ul style="list-style-type: none">• Mejorar la definición de los requerimientos

Fig. 46 Tablas de reunión de análisis del anteproyecto

2.4. Sprint 3

Producto entregable: dentro de la carpeta *source* provista, en el historial, seleccionar el .rar correspondiente a la versión “201505130042” aproximadamente.

2.4.1. Especificación de requerimientos del sprint

RF 1: Página de inicio de sesión

Se deben crear los accesos y registros para usuarios con todo lo que esto conlleva. Este debe ser un punto en el que todo tipo de usuarios pueda acceder con su email y contraseña a los diferentes módulos del sistema.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Utilización de *masterpages* que sean adaptables a dispositivos móviles
- *Responsive design*
- Código HTML y CSS que cumpla con los lineamientos de la W3C (*validators*)
- Validación de atributos (email, contraseña)
- Recuperación de contraseña
- Encriptación de contraseña
- Confirmación de contraseña (repetición)
- Páginas de errores
- Mensaje de bienvenida una vez *logueado*
- Página básica para el cambio de contraseña y gestión de cuenta.

RF2: Acceso a los módulos del sistema

El sistema se debe de poder acceder mediante *urls* que sean descriptivas, creando toda la estructura necesaria (controladores y vistas).

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Priorizar el CEO del sitio para mejorar la indexación de los principales motores de búsqueda. Esto se cumplirá manteniendo un código “limpio” y altamente reutilizable en el cual se utilizará a las clases “*controler*” que redirecciones a las vistas evitando la necesidad de agregar extensiones “.aspx” a las páginas.

RF3: Dos módulos fachada y sistema de gestión integrada

El sistema debe de estar claramente dividido en dos grandes módulos que engloben y categoricen las funcionalidades.

Dichos módulos deben de tener estilos que los diferencien y una forma de presentar las funcionalidades bien delimitadas, a su vez estarán relacionados según el contexto del usuario, con sus respectivas redirecciones.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Utilización de *masterpages* que sean adaptables a dispositivos móviles
- Referencias al nombre de la empresa.
- Una página dentro del módulo de fachada con una pequeña introducción
- Todos deben indicar el lugar que se está situado como título.
- *Dropdowns* y secciones de menú que permitan una fácil adaptación del usuario
- Compatibilidad entre módulos usando una misma tecnología (Razor C#).
- No repetir información entre módulos.

RF4: Gestionar al personal

El sistema debe permitir dar de alta, baja y modificar todo el personal estando organizados según su rol, con determinados permisos y diferentes tareas. Así según su rol podrán acceder a diferentes áreas del sistema.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Los roles serán solo para aquellas entidades que interactúen de forma constante con el sistema, las entidades involucradas en la realización de una obra que no tengan funcionalidades en el sistema no se interpretaran como rol.

RF9: Gestionar los recursos y proveedores.

El sistema debe permitir dar de alta, baja y modificar todos los materiales como “recursos” ya sean maquinaria, herramientas, equipos informáticos, documentos que posea la empresa deben de estar ingresados y organizados. Cada uno de ellos tendrá asociado un proveedor.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Analizar una estructura de “recursos” que englobe a los materiales, herramientas, maquinaria en una estructura abstracta que sea escalable y que abarque la totalidad de los atributos que interesan de cada entidad.
- Cargar dichas estructuras mediante los Excel provistos por el cliente.
- Relacionar los recursos con los proveedores.
- Cargar los proveedores mediante Excel.
- Permitir dar de baja dichas entidades.

- Permitir actualizar dichas entidades.
- Permitir visualizar dichas entidades.
- Permitir listar dichas entidades.
- Permitir dar de alta dichas entidades.
- Utilización de estas acciones en lo posible como del tipo “*in grid*”.
- Ingresar en el menú de gestión integrada los accesos a dichas entidades “*index*”

2.4.2 Conclusiones del sprint

Tareas generales

- Investigación de las mejores prácticas de uso para entity framework
- Versiones experimentales utilizando dicho *framework*
- Primera versión local del proyecto
- Suscripción a Azure
- Deploy de la app local a el entorno de Azure
- Deploy de la local db a una versión en la nube
- Conectar la app a la db
- Actualizar las tablas con Entity Framework
- Utilización de los *application insights* para mantener un monitoreo del sistema
- Verificar mecanismo de *backup* en Azure
- Obtención de FTP para subir de una forma alternativa el sistema (no por deploy) y permitir que el tutor, cliente entre otros puedan verificar el avance interno del sistema.
- Pasaje de un “*directory*” de Azure moviendo todos los servidores y bases de datos al otro *directory*, ya que se logró obtener una nueva suscripción gratuita (BizSpark) gracias a un contacto de trabajo.

RF 1: Página de inicio de sesión

<<Requerimiento cumplido>>

- Utilización de *masterpages* que sean adaptables a dispositivos móviles
 - Se utilizó bootstrap y otras librerías que dieron un resultado limpio y elegante para la presentación del sistema.
- *Responsive design*
 - Se crearon tres plantillas, uno para la gestión integrada, uno para la gestión de usuarios (*login*) y otro moderno e intuitivo para la presentación al cliente de la empresa.
- Código HTML y CSS que cumpla con los lineamientos de la W3C (*validators*)
 - Se aseguró que todo el código de html y css sea el mínimo e indispensable que genere los mejores resultados posibles dado el tiempo que se dispone para realizar este sprint.
- Validación de atributos (email, contraseña)

- Se utilizó e investigó librerías que permitan realizar esto de la mejor forma posible garantizando una compatibilidad con Entity Framework, se utiliza un *checkeo* interno y otro de interfaz, “model.isValid()” y una librería de JavaScript jQuery *validators* respectivamente.
- Recuperación de contraseña
 - Se investigaron las formas más compatibles y rápidas de realizar lo requerido con Entity Framework, donde se optó por utilizar un módulo de gestión de login integrado el cual es altamente escalable pero que llevó un tiempo adaptarse debido a la curva de aprendizaje.
- Encriptación de contraseña
 - Se guardan todas las contraseñas de los usuarios utilizando un encriptado SHA256 en la base de datos con una *key* interna.
- Confirmación de contraseña (repetición)
 - Se aprovechó el sistema de *login* que genera toda la estructura necesaria dando los mensajes al momento de ingresar cada campo.
- Páginas de errores
 - Cada vez que ocurre un error inesperado se crearon *templates* de error con una breve descripción. Falta la utilización de estas ya que de momento existen solo como *templates*.
- Mensaje de bienvenida una vez *logueado*
 - Se muestra un simple mensaje de bienvenida al usuario una vez *logueado*, con la finalidad de ver claramente que usuario entró en sesión.
- Página básica para el cambio de contraseña y gestión de cuenta.
 - Se utilizó parte del sistema de *login* para abarcar lo requerido, el cual solo se muestra cuando un usuario esté *logueado* y se suplanta los menús de registrar y *loguear* por opciones de interés para el usuario *logueado* (*logout*, gestión de cuenta)

RF2: Acceso a los módulos del sistema

<<Requerimiento cumplido>>

- Priorizar el SEO del sitio para mejorar la indexación de los principales motores de búsqueda. Esto se cumplirá manteniendo un código “limpio” y altamente reutilizable en el cual se utilizará a las clases “controller” que redirecciona a las vistas evitando la necesidad de agregar extensiones “.aspx” a las páginas.
 - Para satisfacer este requerimiento se investigó sobre dicha tecnología y se implementó una arquitectura MVC (*model*, *view*, *controller*) que mantiene las capas del sistema separadas y permite la creación de vistas totalmente independientes de código perteneciente al dominio, las cuales fueron creadas utilizando el lenguaje de *markup Razor C#*.

RF3: Dos módulos fachada y sistema de gestión integrada

<<Requerimiento cumplido>>

- Utilización de **masterpages** que sean adaptables a dispositivos móviles
 - Se reutilizo las creadas para el *login* y se implementó de la misma forma en los módulos.
- Referencias al nombre de la empresa.
 - Siempre se visualiza el nombre de la empresa en algún lugar ya sea *footer* o *header*.
- Una página dentro del módulo de fachada con una pequeña introducción.
 - Se creó una página accesible mediante la url base que da una presentación inicial del cliente y del sistema. La cual deriva al inicio de sesión.
- Todos deben indicar el lugar que se está situado como título
 - En el título se muestra dinámicamente el nombre de la página que se está situado.
- *Dropdowns* y secciones de menú que permitan una fácil adaptación del usuario
 - En el módulo de gestión integrada se implementó un *sidemenu* con *dropdowns* que seccionan las funcionalidades del sistema.
- Compatibilidad entre módulos usando una misma tecnología (Razor C#).
 - Todos los módulos tienen la misma arquitectura base, utilizando los mismos principios de programación y nomenclaturas.
- No repetir información entre módulos.
 - Esto se puede asegurar ya que por cada controlador existe una carpeta de vistas y esta carpeta no puede pertenecer a otro controlador.

RF4: Gestionar al personal

<<Requerimiento pendiente>>

- Los roles serán solo para aquellas entidades que interactúen de forma constante con el sistema, las entidades involucradas en la realización de una obra que no tengan funcionalidades en el sistema no se interpretaran como rol.
 - Esta funcionalidad queda pendiente para el sprint 4 ya que se presentaron algunos contratiempos con el re-despliegue del sistema, cuando por motivos económicos se decidió pasar de la suscripción paga a la gratuita conseguida por un contacto con acceso al Spark de Microsoft.

RF9: Gestionar los recursos y proveedores.

<<Requerimiento cumplido>>

- Analizar una estructura de “recursos” que englobe a los materiales, herramientas, maquinaria en una estructura abstracta que sea escalable y que abarque la totalidad de los atributos que interesan de cada entidad.
 - Se creó dicha estructura donde se tiene una entidad llamada “recurso” con una descripción, categoría, unidad, valor, precio, la cual tiene un proveedor y que puede formar parte de cero a muchos componentes.
- Cargar dichas estructuras mediante los Excel provistos por el cliente.
 - Mediante el Excel provisto por el cliente se cargó con datos “genéricos” para comenzar a realizar pruebas.
- Relacionar los recursos con los proveedores.
 - En un principio los recursos serán “genéricos” pero se dejó la posibilidad que el cliente ingrese sus propios recursos y seleccione uno de los proveedores ya establecidos.
- Cargar los proveedores mediante Excel.
 - Se cargó los proveedores mediante un Excel perteneciente al ministerio de transporte y obras públicas. El cual se obtuvo en los principios del sprint 2 cuando se solicitó e investigó por información en diferentes organismos.
- Permitir dar de baja dichas entidades.
 - En los *grids* de cada entidad se puede borrar una entidad que no esté asociada otras entidades.
- Permitir actualizar dichas entidades.
 - Esta acción se encuentra al igual que el *update* sobre el *grid* y permite cambiar los valores de los atributos de la clase.
- Permitir visualizar dichas entidades.
 - Esta acción se encuentra en el *grid* y permite ver en detalle algunos atributos que de otra forma no entrarían en el *grid*.
- Permitir listar dichas entidades.
 - El *grid* lista todas las tuplas de la base de datos.
- Permitir dar de alta dichas entidades.
 - Esta acción se accede mediante un *link* fuera del *grid* que lleva a una página donde se permite dar de alta nuevas instancias de la clase.
- Utilización de estas acciones en lo posible como del tipo “*in grid*”.
 - Dado todo lo anterior se puede asegurar que las acciones que son posibles de implementar en el *grid* fueron ubicadas de forma intuitiva.
- Ingresar en el menú de gestión integrada los accesos a dichas entidades “*index*”.
 - Se dejó pronto para que todas las nuevas funcionalidades del sistema de gestión integrada tengan un sitio en el *sidemenu* agrupándolas por funcionalidad.

2.4.3. Reuniones

2.4.3.1. Devolución del anteproyecto

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/05/07 (2 Horas)	
<i>Número de iteración / sprint</i>	3	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se comentó sobre las posibles formas de implementar lo especificado • Se obtuvo una devolución del anteproyecto • Se obtuvo comentarios sobre la versión experimental que fue creada durante el sprint 2 la cual sirvió para realizar un test funcional inicial del entorno de desarrollo 	<ul style="list-style-type: none"> • No ocurrieron problemas en la iteración, fueron solucionados a tiempo 	<ul style="list-style-type: none"> • <i>Empezar a implementar el diseño de transacciones y las relaciones.</i> • <i>Crear la maqueta inicial</i>

Fig. 47 Tablas de reunión de devolución del anteproyecto

2.5. Sprint 4

Producto entregable: dentro de la carpeta *source* provista, en el historial, seleccionar el .rar correspondiente a la versión “201506031854” aproximadamente.

2.5.1. Especificación de requerimientos del sprint

RF4: Gestionar al personal

Este requerimiento quedó pendiente a implementar en este sprint.

RF5: Gestionar los roles

Para cumplir este requerimiento el sistema debe permitir dar de alta, baja y modificar roles para asignárselo al personal.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión
- Des asignar un rol a un usuario
- Asignar varios roles a un usuario
- Listar roles
- Modificar roles
- Eliminar un rol
- Incorporar roles a la seguridad de los módulos

RF6: Gestionar los clientes

Para cumplir este requerimiento el sistema debe dar de alta, baja y modificar cliente de la empresa constructora y asignarles proyectos.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión
- Listar clientes
- Modificar clientes
- Eliminar un cliente
- Visualizar detalles de un cliente
- Asignarle proyectos a un cliente

RF7: Gestionar proyectos

Un proyecto será una estructura “personal” para cada cliente en donde se podrá generar las estimaciones de precios en base a las obras que contiene. Para cumplir este requerimiento el sistema debe permitir dar de alta, baja y modificar proyectos a los cuales se le asignaran obras.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión
- Listar proyectos
- Modificar proyectos
- Eliminar proyectos
- Visualizar detalles de un proyecto
- Visualizar el cliente de un proyecto
- Asignar un cliente al proyecto

RF8: Gestionar obras

Las obras contendrán los rubros y serán generales para todos los clientes, cumplirán un papel de “template” para que los usuarios las seleccionen y modifiquen según sus necesidades. Para cumplir este requerimiento el sistema debe permitir dar de alta, baja y modificar obras a las cuales se le asignarán componentes que podrán ser materiales y/o personal. Estas podrán ser reutilizadas para los proyectos

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de obras
- Listar obras
- Modificar obras
- Eliminar obras
- Visualizar detalles de una obra
- Asignar un cliente a una obra

2.5.2 Conclusiones del sprint

Tareas generales

- Documentación del sprint
- Diagramas UML de las entidades
- Modificaciones al modelo
- Creación del “wizard” para comenzar a estructurar el cálculo de precios
- Reuniones con el tutor

RF4: Gestionar al personal

<<Requerimiento cumplido>>

RF5: Gestionar los roles

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
 - Se utilizó la *masterpage* de gestión junto a su respectivo controlador. Mediante las etiquetas “[Authorize]” como por ejemplo “[Authorize(Roles="Admin")]” se aplicó seguridad a los módulos.
- Des asignar un rol a un usuario
 - Esto fue cumplido ya que se puede desasignar un rol a un usuario en la sección “roles usuarios”, donde se lista los usuarios y sus respectivos roles permitiendo deseleccionar el rol que se desee desasignar.
- Asignar varios roles a un usuario
 - Fue cumplido, dichos roles aparecen como un control de múltiples “checkboxs” dentro de la vista de usuarios.
- Listar roles
 - Los roles aparecen como un listado en la sección “roles”, por ahora la única información que nos interesa de estos son un identificador y el nombre, con esto ya se puede aplicar la seguridad y accesos a los diferentes controladores.
- Modificar roles
 - Se puede modificar el nombre de un rol, esto generara que se modifiquen para todos los usuarios, sin tener la necesidad de reasignar los mismos ya que se mantiene el id.
- Eliminar un rol
 - Un rol se puede eliminar, lo que repercutiría en los usuarios que lo tengan asignado.
- Incorporar roles a la seguridad de los módulos
 - Como fue mencionado anteriormente utilizar esta forma de organizar los roles, permite que un rol defina los privilegios disponibles para los usuarios asignados a ese rol. Esta propiedad administra los roles con autorización para acceder al controlador o método de acción asociado. Con *tags* del tipo “[Authorize]”.

RF6: Gestionar los clientes

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
 - Se utilizó la *masterpage* de gestión junto a su respectivo controlador. Mediante las etiquetas de seguridad se puede definir la accesibilidad a los módulos.
- Listar clientes
 - Se puede listar los clientes en su respectiva sección
- Modificar clientes
 - En el listado se puede seleccionar un cliente y modificar sus datos
- Eliminar un cliente
 - Se puede eliminar un cliente que no haya tenido proyectos asignados
- Visualizar detalles de un cliente
 - En el listado se puede visualizar los detalles del cliente en otra pantalla
- Asignarle proyectos a un cliente
 - Esto se puede hacer de dos formas, asignarle manualmente en un proyecto al usuario o mediante el *wizard* seleccionar al cliente y crearle un proyecto.

RF7: Gestionar proyectos

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
 - Se utilizó la *masterpage* de gestión junto a su respectivo controlador. Mediante las etiquetas de seguridad se puede definir la accesibilidad a los módulos.
- Listar proyectos
 - Se puede listar los proyectos con sus respectivos clientes asignados
- Modificar proyectos
 - En el listado es posible seleccionar un proyecto y modificarlo
- Eliminar proyectos
 - Así mismo como modificar se permite eliminar un proyecto del listado
- Visualizar detalles de un proyecto
 - Se puede visualizar los detalles de un proyecto con la acción de visualizar en el *grid*
- Visualizar el cliente de un proyecto
 - Este aparecerá en un *combobox* si se accede en la vista de modificar y como texto si se entra a la vista de detalles.
- Asignar un cliente al proyecto
 - En el momento de creación de un proyecto o el de modificación se puede seleccionar una persona del tipo “cliente” y asignarla al proyecto. Esto mismo procedimiento se puede hacer desde el *wizard* siguiendo los pasos correspondientes.

RF8: Gestionar obras

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
 - Se utilizó la *masterpage* de gestión junto a su respectivo controlador. Mediante las etiquetas de seguridad se puede definir la accesibilidad a los módulos
- Listar obras
 - Se puede listar las obras con sus datos básicos
- Modificar obras
 - En el listado es posible seleccionar una obra y modificar sus datos
- Eliminar obras
 - Así mismo como modificar se permite eliminar una del listado
- Visualizar detalles de una obra
 - Se puede visualizar los detalles de una obra con la acción de visualizar en el *grid*
- Asignar una obra a un proyecto
 - En el momento de creación de un proyecto o el de modificación se puede seleccionar una obra y asignarla al proyecto. Esto mismo procedimiento se puede hacer desde el *wizard* siguiendo los pasos correspondientes.

2.5.3. Reuniones

2.5.3.1. Análisis sobre el desarrollo del sistema junto con el tutor

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/05/14 (1 Hora)	
<i>Número de iteración / sprint</i>	4	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se consiguieron diferentes ideas que podrían servir para mejorar las funcionalidades pedidas por el cliente 	<ul style="list-style-type: none"> • Faltó conseguir una solución para conseguir datos de prueba 	<ul style="list-style-type: none"> • Ver si se puede encontrar algún <i>web service</i> o en su defecto consumir los Excel obtenidos durante el sprint 1

Fig. 48 Tablas de reunión de análisis sobre el desarrollo del sistema junto con el tutor

2.5.3.2. Comentarios sobre el desarrollo del sistema

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/05/21 (1 Hora)	
<i>Número de iteración / sprint</i>	4	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se pueden mejorar las funcionalidades pedidas por el cliente 	<ul style="list-style-type: none"> • Faltó crear un diagrama de flujo para mejorar el análisis y desarrollo del problema 	<ul style="list-style-type: none"> • Crear dicho diagrama y continuar con el desarrollo del sistema

Fig. 50 Tablas de reunión de comentarios sobre el desarrollo del sistema

2.5.3.3. Reunión para analizar la idea del “wizard”

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/05/28 (1 Hora)	
<i>Número de iteración / sprint</i>	4	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se obtuvo una forma más dinámica para implementar una interfaz gráfica que funcione como guía para el usuario, que le permita trabajar con el sistema de manera sencilla y amigable 	<ul style="list-style-type: none"> • Faltó definir historias de usuario o <i>mockups</i> para visualizar mejor dicha interfaz 	<ul style="list-style-type: none"> • Elaborar <i>mockups</i> o plantillas HTML que permitan analizar las funcionalidades del <i>wizard</i>

Fig. 51 Tablas de reunión para analizar la idea del “wizard”

2.5.3.4. Reunión donde se cambió de tutor por Fernando Thul

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/06/02 (1 Hora)	
<i>Número de iteración / sprint</i>	4	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Pablo Volpi, Fernando Thul	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Pablo Volpi, Fernando Thul	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se obtuvo nuevas ideas sobre cómo calcular costos para una obra 		<ul style="list-style-type: none"> • Considerar rediseñar algunos aspectos del proceso de estimación de costos acorde a las ideas del tutor

Fig. 52 Tablas de reunión donde se cambió de tutor por Fernando Thul

2.5.4. Reunión de retrospectiva

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Oficina de la empresa IGAFAL S.A	
<i>Fecha</i>	2015/06/03 (1 Hora)	
<i>Número de iteración / sprint</i>	4	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se obtuvo la aprobación del cliente a las nuevas interfaces del sistema • Se corroboró el diagrama de entidades de Entity Framework con el cliente 	<ul style="list-style-type: none"> • Algunos nuevos requerimientos fueron omitidos a futuro, por falta de tiempo 	<ul style="list-style-type: none"> • Ver si se puede encontrar alguna forma de contemplar dichos requerimientos e implementarlos más delante rápidamente

Fig. 53 Tablas de reunión de retrospectiva

2.6. Sprint 5

Producto entregable: dentro de la carpeta *source* provista, en el historial, seleccionar el .rar correspondiente a la versión “201506201503” aproximadamente.

2.6.1. Especificación de requerimientos del sprint

RF10: Gestionar componentes

Para cumplir este requerimiento el sistema debe permitir dar de alta, baja y modificar componentes que serán conformados por recursos, actuarán como contenedores de recursos ya que no tienen una contraparte real, sino que serán utilizados de forma abstracta para facilitar el agrupamiento de recursos que generalmente se ingresan juntos.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión
- Integrar con el *wizard* de rubros
- Listar componentes del rubro
- Listar componentes
- Modificar componentes
- Eliminar un componente
- Visualizar detalles de un componente

RF11: Gestionar estados para las obras

Para cumplir este requerimiento el sistema debe categorizar las obras según su estado donde se podrá distinguir entre obras que estén en planeamiento, desarrollo, finalizadas y en asesoramiento. Esta última se utilizará cuando un cliente solicite asesoramiento para una obra, completando los datos de la solicitud y donde al finalizar el arquitecto le enviará para esa obra en estado de asesoramiento una estimación de costo.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión
- Integrar con el *wizard* de clientes
- Listar las obras según su estado para cada función del sistema según corresponda
- Cambiar el estado de la obra según corresponda en las funciones ya existentes del sistema

RF12: Gestionar relaciones entre componentes

Para cumplir este requerimiento el sistema debe permitir como se mencionó en el RF10 que los componentes estén conformados por otros componentes siendo el tipo de componente más básico los materiales o “recursos” de la lista de precios oficial del ministerio de transporte y obras públicas.

Dado el análisis del requerimiento y reuniones con el cliente, se llegó a la conclusión de que este requerimiento se puede considerar como cumplido si:

- Un componente puede estar formado por uno o más recursos
- Un componente no puede ser editado directamente desde el *wizard* de clientes
- Se puede reutilizar para otros rubros
- Es independiente del metraje de rubros

RF13: Informes sobre el estado de una obra en particular

Para cumplir este requerimiento el sistema debe permitir generar un informe de todos los materiales, maquinaria, herramientas, trabajadores que participen en una obra. Estos deben ser monitoreados para poder generar un informe de estado sobre la obra.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Se debe crear un *wizard* que le facilite al usuario ver el estado general de una obra
- Filtrar por cliente, proyecto, obra y ver toda la información correspondiente
- A su vez en cada paso se deberá poder modificar los datos existentes
- Copiar una obra en base a otra
- Agregar, seleccionar o quitar elementos en cada paso
- Generar graficas usando alguna alternativa *responsive*
- Facilitar la implementación de futuras mejoras al módulo utilizando las mejores prácticas de desarrollo
- Permitir filtrar en cada listado por cualquiera de sus atributos
- Ordenar en cada listado por cualquiera de sus columnas
- Ubicar al usuario dentro del *wizard* con una barra de los pasos previos completados y el actual en curso
- Mensajes simples tipo “notificación” de alerta si ocurren errores inesperados

RF14: Informes de todas las contrataciones públicas/privadas

Para cumplir este requerimiento el sistema debe permitir generar un informe de todas las contrataciones públicas y privadas de la empresa deben ser registradas y ordenadas cronológicamente con los respectivos recursos utilizados para la misma. Por lo tanto, una contratación deberá expandirse con todos los recursos y detalles.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Para aquellas obras que estén finalizadas se permitirá visualizar con que componentes, recursos, costos indirectos, empleados y subcontrataciones fueron realizadas
- Comenzar con la creación de un *wizard* que agrupe las funcionalidades referentes a las obras en desarrollo

RF16: Informes de precios estimados en base al proyecto

Para cumplir este requerimiento el sistema debe permitir que en base al proyecto cuya obra u obras están conformada por componentes se calcule una estimación de precio total. Teniendo en cuenta el beneficio a obtener y otras variables que corresponden al cálculo para la estimación de precios, como por ejemplo los metrajes de rubros, costos indirectos, subcontrataciones entre otros.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Agregar en un historial las vigencias de los recursos al momento de calcular la obra, para facilitar más adelante el monitoreo y reutilizar la obra, recalculando para la fecha actual el precio
- Agrupar todas las entidades que intervienen en el cálculo de precio en una interfaz sencilla que facilite la lectura y comprensión tanto del encargado de calcular la estimación como la de los clientes que recibirán dicha estimación a través de un pedido de asesoramiento

RF23: Cronograma financiero

Para cumplir este requerimiento el sistema debe especificar que para cada fase de la obra una previsión de los gastos. Donde al final se compara el precio estimado inicial con el total del cronograma financiero.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Como punto de partida se espera que el sistema de ese precio estimado inicial y más adelante cuando se desarrollen las funcionalidades referentes al desarrollo de obras realizar la comparación del precio estimado con el real.

RF25: Cronograma de obtención de equipos y materiales

Para cumplir este requerimiento el sistema debe permitir que para cada fase de la obra se registre que materiales se usaron y que equipos se deberían obtener.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Para una fecha dada se debe estimar que recursos y cantidades de deben conseguir, agrupando todos los recursos sin diferenciar entre rubros o componentes a modo de “listado” para facilitar su obtención.

2.6.2 Conclusiones del sprint

Tareas generales

- Documentación del sprint
- Diagramas de secuencia para el funcionamiento del *wizard*
- Modificaciones al modelo
- Cambios al modelo y reorganización de base de datos
- Reuniones con el tutor
- Reuniones con el cliente

RF10: Gestionar componentes

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
 - Se desarrolló esta nueva funcionalidad sobre el módulo de gestión, aplicando la seguridad de dicho modulo. Esto fue realizado a través de un controlador que hereda del controlador del módulo en este caso “ComponentesController” heredando de “GestionController” obliga a que implemente el nivel de acceso base solo a usuarios administrador o empleados.
- Integrar con el *wizard* de rubros
 - Para mejorar la facilidad de acceso a esta funcionalidad se implementó dentro del *wizard* de rubros, donde un operario selecciona un rubro y arma los componentes a medida que va avanzando dentro de los pasos del *wizard*.
- Listar componentes del rubro
 - Al seleccionar un rubro se listan todos los componentes que comprenden dicho rubro permitiendo filtrar y ordenar por cualquiera de sus columnas.
- Listar componentes
 - Además de la forma principal de acceder a los componentes de un rubro se desarrolló una forma de acceder a todos los rubros del sistema sin filtrar por rubros y permitiendo que se pueda visualizar y trabajar con los componentes sin necesidad de estar dentro del contexto de un rubro en el *wizard*.

- Modificar componentes
 - Se puede modificar un componente tanto dentro del *wizard* como fuera del mismo a través de lo mencionado en el punto anterior. Cabe aclarar que solo se podrá armar un componente con sus conjuntos de rubros solo dentro del *wizard* ya que facilita la operatividad del mismo, en el listado general de componentes solo es posible cambiar atributos del propio componente, pero no su contenido (recursos).
- Eliminar un componente
 - Se puede eliminar físicamente un componente desde el listado general o quitar lógicamente un componente del rubro desde el *wizard*.
- Visualizar detalles de un componente
 - Los detalles y recursos de un componente se pueden acceder dentro del *wizard* de rubros y los detalles generales del componente dentro del listado general fuera del *wizard*.

RF11: Gestionar estados para las obras

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
 - Se desarrolló esta nueva funcionalidad sobre el módulo de gestión, aplicando la seguridad de dicho modulo solo a los usuarios que estén *logueados* y que sean administradores y/o empleados.
- Integrar con el *wizard* de clientes
 - Se restructuró el modelo para que solo muestre obras que estén en estado de asesoramiento o planeamiento, debido a que en este módulo se espera que el encargado trabaje con las obras pedidas por clientes que soliciten asesoramiento y dentro del *wizard* de desarrollo estén las obras con los estados correspondientes al desarrollo.
- Listar las obras según su estado para cada función del sistema según corresponda
 - Esto fue implementado en cada lugar donde se referenciaba una obra mostrando las obras que corresponden al contexto y lo que se espera de la funcionalidad.
- Cambiar el estado de la obra según corresponda en las funciones ya existentes del sistema
 - Cuando una obra es calculada se cambia el estado de “asesoramiento” a en “planeamiento” automáticamente.

RF12: Gestionar relaciones entre componentes

<<Requerimiento cumplido>>

- Un componente puede estar formado por uno o más recursos
 - Fue implementado en una tabla que asocia componentes con recursos con correspondencia N a N y a través del *wizard* de rubros es posible modificar y quitar recursos según corresponda.
- Un componente no puede ser editado directamente desde el *wizard* de clientes
 - Fue implementado dentro del *wizard* de rubros ya que continuar con más pasos dentro del mismo, degradaría la usabilidad del módulo. La delimitación es cuando se pide los metrajes de rubros, costos indirectos entre otros y de a partir de allí si se quiere modificar rubros se tendrá que ir al correspondiente *wizard* que engloba todo lo relacionado con la creación y modificación de rubros.
- Se puede reutilizar para otros rubros
 - Un componente es independiente del rubro, se puede utilizar dentro de cualquier rubro.
- Es independiente del metraje de rubros
 - El metraje fue mapeado en una tabla denominada “RubroMetraje” que corresponde al *wizard* clientes, con esto se quiere decir que los rubros no contienen el metraje, sino que los metrajes son para un rubro y ese “RubroMetraje” se guarda en una ICollection dentro de las obras, logrando de esta forma una independencia del rubro con el metraje particular de la obra, permitiendo reutilizar rubros en otras obras.

RF13: Informes sobre el estado de una obra en particular

<<Requerimiento cumplido>>

- Se debe crear un *wizard* que le facilite al usuario ver el estado general de una obra
 - Como se mencionó anteriormente fue creado durante el transcurso de este sprint, donde en los primeros pasos se facilita la creación de las estructuras que contienen la obra y al final se presenta una pantalla con *tabs* relacionados con las entidades que conforman la obra.
- Filtrar por cliente, proyecto, obra y ver toda la información correspondiente
 - Se siguió esta navegabilidad mostrando una barra que informe el paso actual en que se encuentra el usuario dentro del *wizard*.
- A su vez en cada paso se deberá poder modificar los datos existentes
 - En cada paso se da la opción de agregar, eliminar, copiar y seleccionar entidades según corresponda.
- Copiar una obra en base a otra
 - Se creó una vista donde se permite seleccionar una obra existente, copiar sus datos como metrajes de rubros, costos indirectos, sub contrataciones y empleados, y a su vez cambiarle los datos principales como nombre y descripción al momento de copiar.
- Agregar, seleccionar o quitar elementos en cada paso

- Esto fue cumplido según corresponde en cada paso.
- Generar graficas usando alguna alternativa *responsive*
 - Se implementaron las gráficas del sistema utilizando una librería de JavaScript llamada Morris.js que obtiene a través de un evento post los datos *parseados* en un formato JSON de los controladores del sistema y genera las gráficas en html5, js y css dentro de los *views* del cliente.
- Facilitar la implementación de futuras mejoras al módulo utilizando las mejores prácticas de desarrollo
 - Se utilizó una división de tareas clara y altamente escalable, implementando el GRASP patrón experto donde cada capa y clase se ocupa solo para lo que fue creada. El cual es el principio básico de asignación de responsabilidades. Nos indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada favoreciendo la disminución del acoplamiento.
- Permitir filtrar en cada listado por cualquiera de sus atributos
 - Se permite filtrar todas las listas de este módulo por cualquiera de sus atributos.
- Ordenar en cada listado por cualquiera de sus columnas
 - Se permite ordenar todas las listas de este módulo por cualquiera de sus atributos, haciendo *click* sobre los títulos de las columnas para ordenar de forma ascendente o descendente.
- Ubicar al usuario dentro del *wizard* con una barra de los pasos previos completados y el actual en curso
 - Se utilizó un componente de Bootstrap para crear una barra superior que sirva para guiar al usuario a medida que avanza.
- Mensajes simples tipo “notificación” de alerta si ocurren errores inesperados
 - Si, fue implementado en la *masterpage*, estos mensajes se pueden utilizar en cualquier parte que utilice la *masterpage* y se visualizarán en el borde inferior derecho. La forma en que funcionan es sencilla al pasarle un valor de *string* diferente de vacío el mensaje aparecerá, sino este se oculta, de esta forma se pudo lograr de la forma más simple una funcionalidad que aporte a la usabilidad del sistema.

RF14: Informes de todas las contrataciones públicas/privadas

<<Requerimiento cumplido>>

- Para aquellas obras que estén finalizadas se permitirá visualizar con que componentes, recursos, costos indirectos, empleados y subcontrataciones fueron realizadas
 - Este requerimiento fue parcialmente cumplido ya que su prioridad no es alta, pero se dejó una sección pronta para desarrollar más adelante que realizará dicha funcionalidad.
- Comenzar con la creación de un *wizard* que agrupe las funcionalidades referentes a las obras en desarrollo
 - Si, dicho funcionalidad fue denominada “*wizard* desarrollo”.

RF16: Informes de precios estimados en base al proyecto

<<Requerimiento cumplido>>

- Agregar en un historial las vigencias de los recursos al momento de calcular la obra, para facilitar más adelante el monitoreo y reutilizar la obra, recalculando para la fecha actual el precio
 - Si, cada vez que se genera una estimación de costo se guarda un historial con el precio de cada recurso que servirá para monitorear y generar informes de la obra a través del tiempo, así como realizar paramétricas.
- Agrupar todas las entidades que intervienen en el cálculo de precio en una interfaz sencilla que facilite la lectura y comprensión tanto del encargado de calcular la estimación como la de los clientes que recibirán dicha estimación a través de un pedido de asesoramiento
 - Si se integró con el *wizard* existente todas las entidades que intervienen del proceso de estimación de costos. Con esto se logra evitar que el usuario final lea, largos manuales de usuario y aprenda de forma intuitiva en el proceso de que va creando e inicializando las estructuras necesarias para la estimación de costos.

RF23: Cronograma financiero

<<Requerimiento cumplido>>

- Como punto de partida se espera que el sistema de ese precio estimado inicial y más adelante cuando se desarrolle las funcionalidades referentes al desarrollo de obras realizar la comparación del precio estimado con el real.
 - Se desarrolló una vista que al presionar sobre “estimar obra” dentro del *wizard* de clientes, agrupa todos los recursos dentro de los componentes y rubros dando el precio total y generando un informe detallado de las entidades que intervienen en el cálculo.

RF25: Cronograma de obtención de equipos y materiales

<<Requerimiento cumplido>>

- Para una fecha dada se debe estimar que recursos y cantidades de deben conseguir, agrupando todos los recursos sin diferenciar entre rubros o componentes a modo de “listado” para facilitar su obtención.
 - Si, en este sprint el requerimiento fue parcialmente cumplido, más adelante se seguirá desarrollando y agregando información a medida que se desarrolle otras estructuras de las cuales depende y son necesarias para este requerimiento.

2.6.3. Reuniones

2.6.3.1. Análisis sobre el desarrollo del sistema junto con el tutor

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/06/09 (1 Hora)	
<i>Número de iteración / sprint</i>	5	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
• Se consiguió una planilla que sirve como ejemplo para calcular obras		<ul style="list-style-type: none">• Implementar y dar un costo inicial para una obra• Crear un diagrama de secuencia que explique mejor el proceso

Fig. 54 Tablas de reunión de análisis sobre el desarrollo del sistema junto con el tutor

2.6.3.2. Reunión con el tutor para verificar diagrama de secuencia

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/06/16 (1 Hora)	
<i>Número de iteración / sprint</i>	5	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se creó un diagrama de secuencia que sirva para explicar los diferentes flujos que pueden tener los wizard del sistema de gestión integrada • Se corroboró el diagrama de entidades de Entity Framework con el tutor 		<ul style="list-style-type: none"> • Darle prioridad a implementar el diagrama que fue corroborado con el tutor

Fig. 55 Tablas de reunión con el tutor para verificar diagrama de secuencia

2.6.4. Reunión de retrospectiva

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Oficina de la empresa IGAFAL S.A	
<i>Fecha</i>	2015/06/20 (1 Hora)	
<i>Número de iteración / sprint</i>	4	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se obtuvo la aprobación del cliente a las nuevas interfaces del sistema • Se corroboró el diagrama de secuencia que fue creado durante el sprint 	<ul style="list-style-type: none"> • Problema técnico, faltó precompilar las vistas Razor para que cargue más rápido la página, ya que se mostró sin precompilar y demoró un poco más de lo debido, esto no representa la realidad de las capacidades que posee la herramienta utilizada. 	<ul style="list-style-type: none"> • Probar el flujo normal de la aplicación antes de mostrarle al cliente.

Fig. 56 Tablas de reunión de retrospectiva

2.7. Sprint 6

Producto entregable: dentro de la carpeta *source* provista, en el historial, seleccionar el .rar correspondiente a la versión “201507122102” aproximadamente.

2.7.1. Especificación de requerimientos del sprint

RF15: Ingreso de las pautas y criterios para cada proyecto

Para cumplir este requerimiento el sistema debe permitir que para cada proyecto exista un monitoreo sencillo donde se ingrese que pautas y criterios que se requieren para generar la documentación de una obra.

Para ello se debe agregar al *wizard* de rubros una sección que se encuentre preferiblemente al finalizar el rubro, que permita agregar los procedimientos constructivos para dicho rubro, esto sería semejante a una lista de pasos que se deben seguir para completar el proceso que se realiza para desarrollar el rubro. La unión de procedimientos constructivos de los rubros de una obra resultaría en el procedimiento constructivo que se debe seguir “en un principio” para realizar toda la obra.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión
- Integrar con el *wizard* de rubros
- Listar procedimientos del rubro
- Modificar procedimientos del rubro
- Eliminar procedimientos del rubro
- Visualizar detalles de los procedimientos del rubro

RF20: Sección de noticias o novedades sobre la empresa

Para cumplir este requerimiento el sistema debe exponer todas las noticias y/o novedades de la empresa las cuales serán dirigidas al público en general. Las noticias se deben agregar desde el sistema de gestión integrada y ser visualizadas en el módulo de fachada.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Mostrar noticias en la fachada del sitio
- Agregar noticias
- Listar, modificar, eliminar y visualizar procedimientos del rubro

RF26: Instrucciones para el proceso de estimación, desarrollo de obras y otros

Para cumplir este requerimiento el sistema debe mostrar todos los procesos que posee el sistema en instrucciones que expliquen los pasos que se deben seguir para desarrollarlos además de definir y explicar brevemente que realiza cada funcionalidad.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión
- Integrar con el *wizard* de rubros
- Listar instrucciones del rubro
- Modificar instrucciones del rubro
- Eliminar instrucciones del rubro
- Visualizar detalles de las instrucciones del rubro

RF27: Asesoramiento para clientes

Para cumplir este requerimiento el sistema debe permitir que un cliente pida un asesoramiento, especifique que requerimientos quiere de una obra y reciba una estimación de costos. Esto debe estar implementado en el módulo de fachada siendo accesible para usuarios que tengan permiso de cliente, en el cual después de especificar los requerimientos de una obra, envíe el formulario y le avise cuando reciba la estimación esté generada por el sistema de gestión integrada.

Es importante integrar este requerimiento con el módulo de gestión de costos ya existente, aprovechando los *wizard* y agregando las acciones necesarias para trabajar con los asesoramientos de clientes.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar el requerimiento con el módulo de gestión y la fachada
- Crear la estructura necesaria para enviar el formulario de asesoramiento
- Permitir al usuario crear una obra según ese formulario y enviarla al cliente
- Informarle al usuario el estado del asesoramiento y la obra que solicitó
- Mostrar la estimación de solicitada

2.7.2 Conclusiones del sprint

Tareas generales

- Documentación del sprint
- Reestructuración de usuarios
 - Integrar usuarios, roles con empleados y clientes, se tuvo que unificar los diferentes tipos de usuarios heredando de IdentityUser para mejorar la escalabilidad del sistema y manejar para todas las entidades que se logreen al sistema una jerarquía de roles.
- Cambios al modelo y reorganización de base de datos
- Quitar pluralización de las tablas de la base de datos
 - Ya que el nombre de la tabla será el nombre de la clase en plural. Como el servicio de pluralización sólo está disponible en inglés, se desactivo el servicio que pluralizaba las tablas.
- Reuniones con el tutor
- Reuniones con el cliente

RF15: Ingreso de las pautas y criterios para cada proyecto

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
 - Cumplido, se implementó usando la *masterpage* y controlador del módulo de gestión.
- Integrar con el *wizard* de rubros
 - Fue integrado en el *wizard* rubros al final del proceso como un *tab* con listado y que permite agregar procedimientos.
- Listar, modificar, eliminar y visualizar procedimientos del rubro
 - Dichas acciones se pueden acceder cuando se selecciona el *tab*.

RF20: Sección de noticias o novedades sobre la empresa

<<Requerimiento cumplido>>

- Agregar noticias
 - Al agregar una noticia esta se mostrará en la sección de la fachada correspondiente
- Lista noticias
 - A su vez se pueden listar todas las noticias y filtrar.
- Modificar noticias
 - Todas las noticias se pueden modificar desde el *grid*, *cliqueando* en “modificar”
- Eliminar noticias
 - Fue implementado y permite eliminar físicamente las noticias.
- Visualizar noticias
 - Se puede visualizar detalles de la noticia al ingresar a la vista *cliqueando* en “detalles”

RF26: Instrucciones para el proceso de estimación, desarrollo de obras y otros

<<Requerimiento cumplido>>

- Integrar el requerimiento con el módulo de gestión
- Crear la vista con links para proceso de estimación de costos
 - Cumplido, se creó una guía donde se muestra en una serie de pasos como es el flujo normal de estimación de costos.
- Crear la vista con links para otros procesos del sistema
 - Si, se creó una guía donde se describe todas las funcionalidades que no sean del proceso de estimación de costos.

RF27: Asesoramiento para clientes

<<Requerimiento postergado>>

En el siguiente sprint se continuará con el desarrollo de este requerimiento.

- Integrar el requerimiento con el módulo de gestión y la fachada
- Crear la estructura necesaria para enviar el formulario de asesoramiento
- Permitir al usuario crear una obra según ese formulario y enviarla al cliente
- Informarle al usuario el estado del asesoramiento y la obra que solicitó
- Mostrar la estimación de solicitada

2.7.3. Reuniones

2.7.3.1. Reunión con el tutor para informar avances

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/06/23 y 30, 2015/07/14 (1 Hora)	
<i>Número de iteración / sprint</i>	6	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
• Se presentó al tutor los avances realizados durante este sprint		

Fig. 57 Tablas de reunión con el tutor para informar avances

2.7.4. Reunión de retrospectiva

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Oficina de la empresa IGAFAL S.A	
<i>Fecha</i>	2015/07/16 (1 Hora)	
<i>Número de iteración / sprint</i>	6	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se obtuvo la aprobación del cliente a las nuevas interfaces del sistema 	<ul style="list-style-type: none"> • Se especificaron muchos requerimientos nuevos que posiblemente a nivel académico estos no puedan entrar 	

Fig. 58 Tablas de reunión de retrospectiva

2.8. Sprint 7

Producto entregable: dentro de la carpeta *source* provista, en el historial, seleccionar el .rar correspondiente a la versión “201508032242” aproximadamente.

2.8.1. Especificación de requerimientos del sprint

RF27: Asesoramiento para clientes (continuación)

Para continuar con el desarrollo de éste requerimiento se deberá permitir que un cliente pueda pedir un asesoramiento desde una interfaz en el módulo de fachada, crear un instructivo que motive al cliente a utilizarlo y le explique a medida que continua con el proceso lo que sucede y sucederá en cada paso.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Interfaz de instrucciones
- Interfaz de “pedido de asesoramiento”
- Estados para un proyecto
- Dividir en áreas de MVC los módulos
- Integrar asesoramientos con el *wizard* para clientes

RF28: Perfil para cliente con tareas (nuevo)

Para cumplir con éste requerimiento el sistema deberá tener una interfaz con el perfil del usuario *logueado*, donde se pueda visualizar todas las tareas correspondientes al rol “cliente”.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Diseño *responsive*
- Imagen de usuario
- Datos principales
- Incluir tareas principales
- Configuración de perfil

RF29: Notificaciones (nuevo)

Para cumplir con éste requerimiento el sistema debe generar notificaciones para las distintas ocurrencias del sistema, las cuales servirán para que el usuario del sistema conozca en detalle todo lo que ocurre en el mismo.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Generar notificaciones
- Entidades y enumerados para notificaciones
- Integrar con *masterpage*
- Tiempo transcurrido desde la notificación

2.8.2 Conclusiones del sprint

Tareas generales

- Documentación del sprint
- Modificaciones al modelo
- Cambios al modelo y reorganización de base de datos
- Reuniones con el tutor
- Reuniones con el cliente

RF27: Asesoramiento para clientes (continuación)

<<Requerimiento cumplido>>

- Interfaz de instrucciones
 - Se crearon la interfaz de asesoramiento de obras, interfaz de instrucciones y consultas de asesoramiento para clientes y se probó enviar un post del *form*. Además, se crearon las consultas de asesoramiento, ver estados y proyectos estimados para cliente, con una sección en donde se concentrará las tareas que podrá realizar el cliente y así mismo ver su perfil.
- Interfaz de “pedido de asesoramiento”
 - Se envía un “pedido” de asesoramiento al *wizard* de clientes, donde se inicializará un proyecto donde se podrá crear y asignar las obras que correspondan a las especificaciones que pidió el cliente en el momento de crear el asesoramiento.
- Estados para un proyecto
 - Para poder diferenciar los diferentes estados de los proyectos se creó y asignó un enumerado con 4 estados: estimando para cuando recién se crea el pedido de asesoramiento, estimado para cuando se finaliza y calcula una estimación para el proyecto, construyendo para cuando el proyecto se concreta y se empieza la fase de desarrollo y por último, construido que se asigna cuando todas las obras del proyecto quedan finalizadas.
- Dividir en áreas de MVC los módulos
 - Para poder dividir los controladores y vistas del sistema, se tuvo que crear y configurar las áreas, reubicando en cada área los controladores que correspondan y pasar todos los links a Razor *links/actions*, actualizar el *route config* y modificar todas las navegaciones del sistema para que correspondan a la nueva disposición.
- Integrar asesoramientos con el *wizard* para clientes
 - El cliente envía una consulta y en gestión se debe recibir y acoplar el asesoramiento con el *wizard* de clientes para dar una estimación de costo seleccionando uno de los pedidos de asesoramiento del cliente.

RF28: Notificaciones (nuevo)

<<Requerimiento cumplido>>

- Generar notificaciones
 - Para aquellas funcionalidades que interesa que se registre una notificación, se agregó la lógica para crear una notificación, algunas de esas funcionalidades son: registro de usuarios, pedidos de asesoramiento, inicio de sesión entre otros. Cabe aclarar que esta funcionalidad se programó de forma tal que quitar o agregar nuevos estados es sencillo y adaptable a nuevos requerimientos.
- Entidades y enumerados para notificaciones
 - Se creó la entidad “notificación” con sus respectivos enumerados que indicarán el estado de cada notificación.
- Integrar con *masterpage*
 - Para que el usuario del sistema de gestión integrada pueda visualizar de manera sencilla las notificaciones, se creó un *dropdown* ordenado por fecha para las últimas 5 notificaciones.
- Tiempo transcurrido desde la notificación
 - Se agregó métodos que permitan saber el tiempo transcurrido desde cada notificación y se integró con el *dropdown* de la *masterpage*.

Tareas adicionales

- Tarea 4 - Test y correcciones que se fueron encontrando.
- Tarea 5 - Reunión con el cliente

Se estableció que el enfoque del proyecto sea en pedidos de asesoramiento y estimación de costos, pero permitiendo la escalabilidad a futuros nuevos requerimientos.
- Tarea 6 - Análisis para vigencias de recursos e historial de obras
- Tarea 7 - Vigencias para obras

Se agregó vigencias para obras, un *backup* de la base de datos y mejoras para la estimación de precios.
Grafica lineal para estimaciones y fechas, *fix* del tema en la *masterpage* gestión, *navbar responsive* y actualización de instrucciones.

2.8.3. Reuniones

2.8.3.1. Reunión con el tutor para informar avances

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/07/21 y 28, 2015/08/04 (1 Hora)	
<i>Número de iteración / sprint</i>	7	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
• Se presentó al tutor los avances realizados durante este sprint		

Fig. 59 Tablas de reunión con el tutor para informar avances

2.8.4. Reunión de retrospectiva

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Oficina de la empresa IGAFAL S.A	
<i>Fecha</i>	2015/08/03 (1 Hora)	
<i>Número de iteración / sprint</i>	7	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se presentó lo realizado en el transcurso del sprint. 		<ul style="list-style-type: none"> • Algunas funcionalidades cambiaron y se tuvo que reorganizar las tareas para el sprint 7 y 8

Fig. 60 Tablas de reunión de retrospectiva

2.9. Sprint 8

Producto entregable: dentro de la carpeta *source* provista, en el historial, seleccionar el .rar correspondiente a la última versión de la carpeta.

2.9.1. Especificación de requerimientos del sprint

RF18: Prevenir y gestionar incidentes (modificado)

El sistema deberá permitir que tanto los clientes como los administradores del sistema puedan reportar todo lo que sucede durante la construcción de las obras. Ya sean problemas, información adicional, ocurrencias u otro tipo de eventos. Cabe aclarar que este requerimiento fue modificado de lo descripto en el anteproyecto para que se ajuste a las nuevas necesidades del cliente.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar con el módulo de fachada
- Integrar con las tareas del cliente
- Integrar con el *wizard* de clientes
- Crear *prompts* para los detalles de los incidentes

RF20: Sección de noticias o novedades sobre la empresa

El sistema debe exponer todas las noticias y/o novedades de la empresa las cuales serán dirigidas al público en general.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar con el módulo de fachada
- Integrar con el módulo de sistema de gestión integrada
- Crear contenidos

RF21: Reportar un incidente (modificado)

El sistema debe permitirle a cualquier miembro del personal a reportar un accidente. Dada la nueva especificación de este requerimiento, éste funcionará como una forma de comunicar lo que ocurre en el transcurso de las obras.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Listar incidentes de manera intuitiva
- Cliente reporta incidentes.

RF22: Tiempo sin incidentes (modificado)

El sistema debe mostrar los incidentes ordenados por fecha y notificar en un *dropdown* separado al administrador, para que el mismo pueda gestionar y tomar las medidas necesarias más fácilmente.

Dado el análisis del requerimiento y reuniones con el cliente, se deberán de tener en cuenta los siguientes puntos:

- Integrar con el módulo de gestión integrada

2.9.2 Conclusiones del sprint

Tareas generales

- Documentación del sprint
- Modificaciones al modelo
- Cambios al modelo y reorganización de base de datos
- Reuniones con el tutor
- Reuniones con el cliente
- *Testing* general
- Limpieza de código
- Contactar a soporte de Twilio

RF18: Prevenir y gestionar incidentes (modificado)

<<Requerimiento cumplido>>

- Integrar con el módulo de fachada
 - Se crearon los controladores y vistas que correspondan a la visualización y acciones que correspondan al rol de cliente.
- Integrar con las tareas del cliente
 - Se integró la interfaz correspondiente, dentro del perfil en la tarea “construcciones”.
- Integrar con el *wizard* de clientes
 - Dentro del *wizard* cuando se llega a la etapa de personalizar una obra, se puede trabajar con los incidentes, asignando y modificando incidentes.
- Crear *prompts* para los detalles de los incidentes
 - Si un usuario selecciona un incidente, haciendo *click* en la fecha se desplegará un *prompt* con los detalles.

RF20: Sección de noticias o novedades sobre la empresa

<<Requerimiento cumplido>>

- Integrar con el módulo de fachada
 - Se creó una sección que despliega “tarjetas” con todas las noticias y novedades. Estas fueron diseñadas de manera responsiva y se reorganizan según el ancho de su contenido acomodándose a los distintos tamaños de las pantallas.
- Integrar con el módulo de sistema de gestión integrada
 - Para agregar nuevas noticias se debe ingresar al sistema de gestión integrada y en la sección de noticias se puede gestionar el contenido.
- Crear contenidos
 - Se creó contenido de ejemplo publicitando las nuevas funcionalidades del sistema.

RF21: Reportar un incidente (modificado)

<<Requerimiento cumplido>>

- Listar incidentes de manera intuitiva
 - Se listan los incidentes para todas las obras del proyecto como paneles con *grids*, donde se puede *clickear* en la fecha para ver detalles.
- Cliente reporta incidentes.
 - Un cliente puede reportar un incidente, especificando fecha, una descripción y seleccionando alguna de las obras que posea.

RF22: Tiempo sin incidentes (modificado)

<<Requerimiento cumplido>>

- Integrar con el módulo de gestión integrada
 - Se creó un *dropdown* en el módulo de gestión integrada, con los últimos incidentes y con el correspondiente tiempo que transcurrió desde cada incidente.

Tareas adicionales

- Tarea 1 – Mostrar en el *wizard* los asesoramientos y avisar por notificación
Se está analizando cómo y dónde integrar los asesoramientos para visualizar y completar las obras pedidas según corresponda.
Se implementó parte de la interfaz y métodos para los controladores.
- Tarea 2 – Enviarle al cliente las estimaciones de proyectos
Según si el administrador calcula el proyecto, cambiará el estado del proyecto y el cliente podrá visualizar la estimación.
- Tarea 3 – Integrar sistema con envío de emails
Se envían mails para confirmar el email de registro, resetear la contraseña, verificación en dos pasos y con las notificaciones del sistema.
- Tarea 4 – *Tabs* de obras para un proyecto y cronograma de obtención de recursos
Se implementó *tabs* para obras dentro del proyecto del cliente y mejoras en el cronograma de obtención de recursos.
Se aplicó patrón experto para facilitar el cálculo de estimaciones y el cronograma de obtención de recursos.
- Tarea 5 – Integrar envío de sms
Para verificación en dos pasos y notificaciones del sistema
- Tarea 6 – Descripciones y auto documentación de interfaces
Descripciones a cada interfaz para guiar al usuario.
- Tarea 7 – Mail al pedir asesoramiento
Se envía mail a todos los *admins* del sistema notificando el nuevo pedido de asesoramiento.
- Tarea 8 – Descripciones y documentación de *wizards*
- Tarea 9 – *Timeline* para procesos constructivos
Se integró un *timeline* para los procesos constructivos, *testing* general y modificaciones varias.
- Tarea 10 – *Testing* y limpieza de código
Testing, optimización, limpieza y actualización de *packages* NuGet.
- Tarea 11 – Pasos para pedido de asesoramiento
Dividir la sección de pedidos de asesoramiento en pasos que mejoren y faciliten la organización.
- Tarea 12 – Plan para proyectos de clientes
Presentarle al cliente mediante el módulo de fachada los procedimientos constructivos generados como plan de proyecto para las obras.
- Tarea 13 – Incidentes para obras
Se implementaron los incidentes para obras dentro del *wizard* y el *dropdown* del menú superior en el módulo de gestión.
- Tarea 14 – Nómina de contrataciones público-privada e incidentes para obras
Se creó una sección denominada nómina de contrataciones público- privada con los proyectos construidos y se siguió con la implementación de incidentes para obras.
- Tarea 15 – Documentación y últimos arreglos

Uno de los contactos a soporte de Twilio

Alrededor de los días 11 y 13 de agosto se contactó al servicio de envío de sms y notificaciones “Twilio” por un tema técnico con el envío de sms desde un número estadounidense toll free provisto por ellos hacia un número local Ancel.

El problema ocurría cuando se enviaba una notificación del sistema como, por ejemplo, un código de verificación en dos pasos y este no llegaba a destino.

El mismo no era causado ni por el código fuente del proyecto ni por la API, ya que este usaba la API correctamente y se podía verificar en los *logs* del *dashboard* de Twilio, en donde se cobraba cada mensaje que se generaba desde el *backend* y se marcaba como enviado a destino posteriormente. Esto era un falso positivo ya que lo último no ocurría, porque Twilio los enviaba, pero Ancel los filtraba, siendo esto causado por el distribuidor de la operadora, en este caso Antel redistribuía los mensajes con un numero ej. 7020 y aplicaba un filtro que causaba la eliminación del mensaje.

Esto se pudo solucionar contactando con el servicio y hablando por mail y telefónicamente hasta solucionar aparentemente el problema.

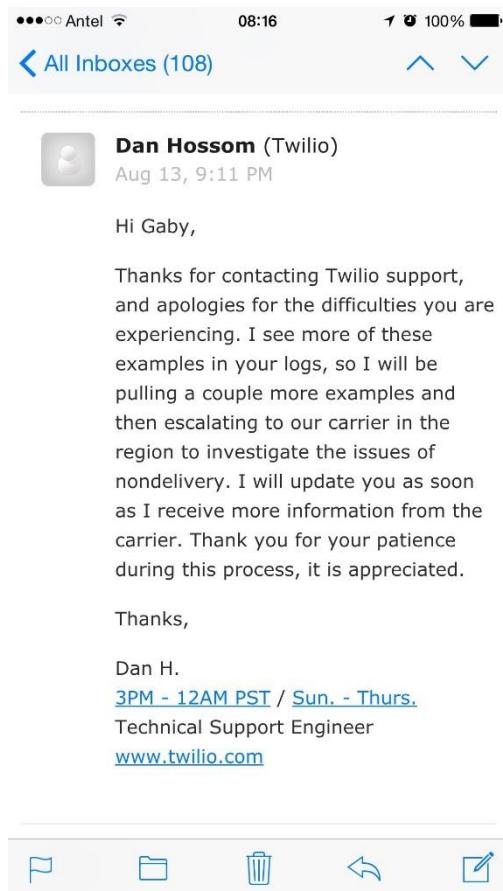


Fig. 61 Screenshot de contacto con soporte de Twilio

2.9.3. Reuniones

2.9.3.1. Reunión con el tutor para informar avances

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Universidad ORT	
<i>Fecha</i>	2015/08/11 (1 Hora)	
<i>Número de iteración / sprint</i>	8	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Fernando Thul	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
• Se presentó al tutor los avances realizados durante este sprint		

Fig. 62 Tablas de reunión con el tutor para informar avances

2.9.4. Reunión de retrospectiva

Empresa Organización IGAFAL S.A

<i>Proyecto</i>	Sistema de gestión de obras	
<i>Lugar</i>	Oficina de la empresa IGAFAL S.A	
<i>Fecha</i>	2015/08/15 (4 Horas)	
<i>Número de iteración / sprint</i>	8	
<i>Personas convocadas a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>Personas que asistieron a la reunión</i>	Gabriel Martínez, Jorge Carámbula Igarabide	
<i>¿Qué salió bien en la iteración? (aciertos)</i>	<i>¿Qué no salió bien en la iteración? (errores)</i>	<i>¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua)</i>
<ul style="list-style-type: none"> • Se presentó lo realizado en el transcurso del sprint. 		

Fig. 63 Tablas de reunión de retrospectiva

2.10. Tabla de cumplimiento de requerimientos

<i>Nombre de requerimiento</i>	<i>Nº de requerimiento</i>	<i>Cumplido</i>
Página de inicio de sesión	RF1	Si (sprint 3)
Acceso a los módulos del sistema	RF2	Si (sprint 3)
Dos módulos fachada y sistema de gestión integrada	RF3	Si (sprint 3)
Gestionar al personal	RF4	Si (sprint 4)*
Gestionar los roles	RF5	Si (sprint 4)
Gestionar los clientes	RF6	Si (sprint 4)
Gestionar los proyectos	RF7	Si (sprint 4)
Gestionar las obras	RF8	Si (sprint 4)
Gestionar recursos y proveedores	RF9	Si (sprint 3)
Gestionar componentes	RF10	Si (sprint 5)
Gestionar estados para las obras	RF11	Si (sprint 5)
Gestionar relaciones entre componentes	RF12	Si (sprint 5)
Informes sobre el estado de una obra en particular	RF13	Si (sprint 5)
Informes de todas las contrataciones públicas/privadas	RF14	Si (sprint 5)
Ingreso de las pautas y criterios para cada proyecto	RF15	Si (sprint 6)
Informes de precios estimados en base al proyecto	RF16	Si (sprint 5)
Exponer cargos disponibles en la empresa.	RF17	Baja prioridad
Prevenir y gestionar accidentes	RF18	Si (sprint 8)
Calcular la paramétrica para una obra que no sea de este año	RF19	Baja prioridad
Sección de noticias o novedades sobre la empresa	RF20	Si (sprint 6)
Reportar un accidente	RF21	Si (sprint 8)
Tiempo sin accidentes	RF22	Si (sprint 8)
Cronograma financiero	RF23	Si (sprint 5)
Cronograma físico	RF24	Baja prioridad
Cronograma de obtención de equipos y materiales	RF25	Si (sprint 5)
Instrucciones para el proceso de estimación, desarrollo de obras y otros	RF26	Si (sprint 6)
Asesoramiento para clientes	RF27	Si (sprint 6)

* Este requerimiento quedó pendiente y se implementó en este sprint.

Fig. 64 Tablas de cumplimiento de requerimientos

2.11. Funcionalidades del producto, casos de uso

A continuación, se dará ejemplos de casos de uso para organizar las distintas funcionalidades del producto que fueron listadas brevemente en el anteproyecto.

Se listan a modo de ejemplo ya que el foco del proyecto fue el desarrollo en base a las tareas de documentación más relevantes para satisfacer las necesidades del cliente.

Creación de usuario

Nombre:	Creación de usuario
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción:	
Permite crear un nuevo usuario del sistema. Como este proyecto es privado no se permitirá el ingreso de nuevos usuarios públicamente, sino que será realizado por un administrador, este completará los datos requeridos como la contraseña y el email.	
Por lo tanto, los tipos de usuario creables inicialmente son:	
<ul style="list-style-type: none">• Administrador• Arquitecto• Otros creados “manualmente” (Por caso de uso “Creación de rol”).	
Actores:	
<ul style="list-style-type: none">• Administrador	
Precondiciones:	
<ul style="list-style-type: none">• El actor debe haberse <i>logueado</i> en el sistema.• El actor debe conocer todos los datos del usuario a crear.	
Flujo Normal:	
<ol style="list-style-type: none">1) Actor entra al sitio.2) Actor ingresa al módulo de gestión e ingresa a “registrar”.3) Actor completa los datos requeridos.4) Actor presiona en crear el nuevo usuario.5) Sistema autoriza el nuevo usuario.6) Sistema re direcciona al usuario al <i>dashboard</i>.7) Sistema notifica el resultado al actor.	
Flujo Alternativo:	
<ol style="list-style-type: none">6) Sistema no autoriza el nuevo usuario.<ol style="list-style-type: none">a) Sistema indica los campos con error.b) Actor corrige los errores.c) Actor continúa desde el paso 3 del flujo normal.	
Postcondiciones:	
<ul style="list-style-type: none">• El usuario fue creado y notificado.	

Fig. 65 Tabla de caso de uso creación de usuario

Modificación de rol para un usuario

Nombre:	Modificación de rol para un usuario
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción:	
Permite asignar un nuevo rol a un usuario ya existente en el sistema, dicho rol le otorgara nuevos permisos al usuario.	
Actores:	
<ul style="list-style-type: none">• Administrador	
Precondiciones:	
<ul style="list-style-type: none">• El actor debe haberse <i>logeado</i> en el sistema.	
Flujo Normal:	
<ol style="list-style-type: none">1) Actor entra al módulo de asignación de roles.2) Actor selecciona el usuario por búsqueda o lista.3) Actor selecciona un nuevo rol de los existentes menos el ya asignado.4) Actor presiona en asignar rol.5) Sistema corrobora los datos.6) Sistema notifica resultado al actor.	
Flujo Alternativo:	
<ol style="list-style-type: none">5) Sistema corrobora los datos y son inválidos o nulos.<ol style="list-style-type: none">a) Sistema indica los campos con error.b) Administrador corrige los errores.c) Actor continúa desde el paso 4 del flujo normal.	
Postcondiciones:	
<ul style="list-style-type: none">• El rol del usuario fue modificado y los nuevos permisos son asignados.	

Fig. 66 Modificación de rol para un usuario

Creación de rol

Nombre:	Creación de rol
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción: Permite crear un nuevo rol en sistema. Ingresando nombre de rol y una descripción.	
Actores: <ul style="list-style-type: none">• Administrador	
Precondiciones: <ul style="list-style-type: none">• El actor debe haberse <i>logueado</i> en el sistema.	
Flujo Normal: <ol style="list-style-type: none">1) Actor entra al módulo de creación de roles.2) Actor completa los siguientes datos.<ol style="list-style-type: none">a) Nombre de rolb) Descripción3) Actor presiona en crear el nuevo rol.4) Sistema autoriza el nuevo rol.5) Sistema notifica resultado al actor.	
Flujo Alternativo: <ol style="list-style-type: none">3) Sistema no autoriza el nuevo el nuevo rol.<ol style="list-style-type: none">a) Sistema indica los campos con error.b) Actor corrige los errores.c) Actor continúa desde el paso 5 del flujo normal.	
Postcondiciones: <ul style="list-style-type: none">• El nombre de rol fue creado con éxito.	

Fig. 67 Tabla de caso de uso creación de rol

Modificación de estado para un usuario (inactivar)

Nombre:	Modificación de estado para un usuario (inactivar)
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción:	Permite inactivar a un usuario, prohibiéndole el ingreso a todo el sistema. Y enviándole el motivo.
Actores:	<ul style="list-style-type: none"> • Administrador
Precondiciones:	<ul style="list-style-type: none"> • El actor debe haberse <i>logeado</i> en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1) Actor entra al módulo de bloqueos de roles. 2) Actor selecciona el usuario por lista filtrada. 3) Actor selecciona o desSelecciona el usuario a bloquear/desbloquear. 4) Actor completa el motivo para el bloqueo o desbloqueo. 5) Actor presiona en efectuar cambios. 6) Sistema corrobora los datos. 7) Sistema notificara vía email al usuario que su estado fue modificado por actor. 8) Sistema notifica resultado al actor.
Flujo Alternativo:	<ol style="list-style-type: none"> 6) Sistema corrobora los datos y son inválidos o nulos. <ol style="list-style-type: none"> a) Sistema indica los campos con error. b) Administrador corrige los errores. c) Actor continúa desde el paso 5 del flujo normal.
Postcondiciones:	<ul style="list-style-type: none"> • El estado del usuario fue modificado con éxito.

Fig. 68 Tabla de caso de uso modificación de estado para un usuario (inactivar)

Generación de listado de todos los usuarios (filtrados)

Nombre:	Generación de listado de todos los usuarios (filtrados)
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción:	
Permite al actor ver una lista de todos los usuarios del sistema.	
Actores:	
<ul style="list-style-type: none">• Administrador	
Precondiciones:	
<ul style="list-style-type: none">• El actor debe haberse <i>logeado</i> en el sistema.	
Flujo Normal:	
<ol style="list-style-type: none">1) Actor entra al módulo de listado de usuarios.2) Actor selecciona el tipo de filtrado.<ol style="list-style-type: none">a) Segundo rol.b) Sin filtrar (todos).3) Sistema genera una lista con todos los datos de los usuarios filtrados.	
Flujo Alternativo:	
<ol style="list-style-type: none">3) Sistema no genera lista de usuarios<ol style="list-style-type: none">a) Sistema indica que no existe usuarios con ese filtro o no existen.b) Actor corrige los errores.c) Actor continúa desde el paso 2 del flujo normal.	
Postcondiciones:	
<ul style="list-style-type: none">• La lista de usuarios fue generada con éxito.	

Fig. 69 Tabla de caso de uso generación de listado de todos los usuarios (filtrados)

Creación de proyecto

Nombre:	Creación de proyecto
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción:	
Permite crear un nuevo proyecto en el sistema. Será realizado por un administrador, este completará los datos requeridos y se le asignará un usuario. Estos proyectos son específicos para un cliente y el cliente podrá tener varios proyectos asignados.	
Actores:	
<ul style="list-style-type: none"> • Administrador 	
Precondiciones:	
<ul style="list-style-type: none"> • El actor debe haberse <i>logeado</i> en el sistema. • El actor debe conocer todos los datos del proyecto a crear. 	
Flujo Normal:	
<ol style="list-style-type: none"> 1) Actor entra al sitio. 2) Actor ingresa al módulo de gestión e ingresa a “Proyectos”. 3) Ingresar en “Crear nuevo” 4) Actor completa los datos requeridos. <ol style="list-style-type: none"> a) Nombre b) Fecha c) País d) Costo e) ClientId (Seleccionando desde un <i>combobox</i> el nombre) 5) Actor presiona en crear el nuevo proyecto. 6) Sistema autoriza el nuevo proyecto. 7) Sistema redirecciona al usuario a los detalles del proyecto. 8) Sistema notifica el resultado al actor. 	
Flujo Alternativo:	
<ol style="list-style-type: none"> 1) Sistema no autoriza el nuevo proyecto. <ol style="list-style-type: none"> a) Sistema indica los campos con error. b) Actor corrige los errores. c) Actor continúa desde el paso 4 del flujo normal. 	
Postcondiciones:	
<ul style="list-style-type: none"> • El proyecto fue creado y asignado a un cliente. 	

Fig. 70 Tabla de caso de uso creación de proyecto

Creación de obras

Nombre:	Creación de obras
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción:	
Permite crear una nueva obra en el sistema. Será realizado por un administrador, este completará los datos requeridos. Estas obras serán como “ <i>templates</i> ” genéricos las cuales se podrán seleccionar y posteriormente contendrán toda la estructura de rubros y componentes.	
Actores:	
<ul style="list-style-type: none">• Administrador	
Precondiciones:	
<ul style="list-style-type: none">• El actor debe haberse <i>logeado</i> en el sistema.• El actor debe conocer todos los datos de la obra a crear.	
Flujo Normal:	
<ol style="list-style-type: none">1) Actor entra al sitio.2) Actor ingresa al módulo de gestión e ingresa a “Obras”.3) Ingresar en “Crear nuevo”4) Actor completa los datos requeridos.<ol style="list-style-type: none">a) Nombreb) Fechac) Imagend) Costo5) Actor presiona en crear nueva obra.6) Sistema autoriza la nueva obra.7) Sistema re direcciona al usuario a los detalles de la obra.8) Sistema notifica el resultado al actor.	
Flujo Alternativo:	
<ol style="list-style-type: none">1) Sistema no autoriza la nueva obra.<ol style="list-style-type: none">a) Sistema indica los campos con error.b) Actor corrige los errores.c) Actor continúa desde el paso 4 del flujo normal.	
Postcondiciones:	
<ul style="list-style-type: none">• La obra fue creada.	

Fig. 71 Tabla de caso de uso creación de obras

Creación de cargos

Nombre:	Creación de cargos
Autor:	Gabriel Martínez
Fecha:	02/05/2015
Descripción:	
Permite crear una nuevo cargo en el sistema. Será realizado por un administrador, este completará los datos requeridos. Estos cargos serán asignados a una categoría y posteriormente a un personal.	
Actores:	
<ul style="list-style-type: none">• Administrador	
Precondiciones:	
<ul style="list-style-type: none">• El actor debe haberse logeado en el sistema.• El actor debe conocer todos los datos de la obra a crear.	
Flujo Normal:	
<ol style="list-style-type: none">1) Actor entra al sitio.2) Actor ingresa al módulo de gestión e ingresa a “Cargos”.3) Ingresar en “Crear nuevo”4) Actor completa los datos requeridos.<ol style="list-style-type: none">a) Nombreb) Descripción5) Actor presiona en crear nuevo cargo.6) Sistema autoriza el nuevo cargo.7) Sistema re direcciona al usuario a los detalles del cargo.8) Sistema notifica el resultado al actor.	
Flujo Alternativo:	
<ol style="list-style-type: none">1) Sistema no autoriza la nueva obra.<ol style="list-style-type: none">a) Sistema indica los campos con error.b) Actor corrige los errores.c) Actor continúa desde el paso 4 del flujo normal.	
Postcondiciones:	
<ul style="list-style-type: none">• El cargo fue creado.	

Fig. 72 Tabla de caso de uso creación de cargos

2.12. Otros requerimientos no funcionales

Requerimientos de performance

Se requiere que el sistema sea eficiente al momento de utilizar los recursos de la base de datos, debido a que estos son limitados. Por esto se utilizará un patrón arquitectónico que cumpla los requisitos y se adapte lo mejor posible a la realidad del sistema.

Las consultas no deberían de superar los 5 segundos como máximo para no interferir la eficiencia de los actores del sistema en el cumplimiento de sus tareas.

Aparte se debe tener en cuenta el poco ancho de banda a internet de los usuarios (30Mbps), la capa de presentación no debe ser exigente ni en procedimientos (pasos) o en recursos (descarga o subido).

Requerimientos de seguridad física (safety)

Se debe cumplir con una buena seguridad ya que este sistema es vital para el funcionamiento de la empresa constructora, por lo tanto, se debe proteger las entradas y salidas del sistema con un firewall.

Se recomienda mantener un *backup* automático de la base de datos y tener una redundancia de datos por posibles fallas electrónicas. No se recomienda dar acceso *root* a el servidor a nadie después de instalado el sistema.

Requerimientos de seguridad lógica (security)

Se debe mantener la privacidad de los usuarios y estos no deben interferir con las áreas de otros. Por lo tanto, se tiene que cifrar todos los datos que sean sensibles para prevenir posibles accesos no autorizados ya sea dentro o fuera de la empresa.

Las contraseñas de cada actor del sistema deben ser fuertes y nunca ser revelada a otro, ni siquiera a los administradores.

Atributos de calidad de software

El sistema tiene que priorizar la facilidad de uso y adaptarse fácilmente a lo que el usuario espera, no debe de realizar funcionalidades que no se requieran, manteniendo una organización sencilla y clara. Se requiere una solución robusta que realice lo que se pide y nada más. Por lo tanto, debe ser confiable al momento de operar.

Por otro lado, el mantenimiento del mismo se tiene que reducir y automatizar lo más posible.

Reglas del negocio

Se espera que una autoridad externa a la empresa constructora pueda acceder al sistema. Teniendo derecho a inspeccionar la base de datos si es necesario. Esto solo ocurrir si la inspección tiene carácter legal obligatorio.

2.13. Requisitos para la interfaz externa

Interfaces de Usuario

Como interfaz de usuario se utilizará HTML, CSS y JS desde cualquier navegador que lo soporte. Implementando una interfaz sencilla y fácil de entender para cualquier usuario.

Ejemplos interfaz de inicio y de login de usuarios (en las primeras versiones del sistema):

SOBRE IGAFAL SRL

Brindamos proyectos y construcciones a clientes e inversores nacionales e internacionales dentro de cualquier ramo de la arquitectura.

Nuestros proyectos pueden ser tanto públicos como privados siguiendo los más altos estándares constructivos.

No preocúpate por la imagen empresarial y asegurarnos garantías intentando mejorar puntualmente los beneficios obtenidos y la seguridad de nuestros empleados.

INGRESAR AL SITIO

Te registras en IGAFAL SRL, trae los materiales necesarios para que te pague y te integre como empleado o proveedor de servicios de IGAFAL SRL. Una vez que seas empleado de IGAFAL SRL.

CONTACTANOS

Síntetizate libre de contactarnos ante cualquier duda, problema, sugerencia... Simplemente envíanos un correo con tu cuestión a nuestra cuenta de correo electrónico y te responderemos a la brevedad

info@igafal.com

[TWITTER](#) [LINKEDIN](#) [GOOGLE+](#)

Copyright © 2015 IGAFAL by Getty RM

Nombre	Descripción	Unidad	Valor	Costo	Tipo
Generico	Ventana corrediza 140x110	U	1	2600.00	Aberfuradasequipamientos
Generico	Puerta ventana 140x205	U	1	5230.00	Aberfuradasequipamientos
Generico	Puerta de calle con postigo 83x210	U	1	5400.00	Aberfuradasequipamientos
Generico	Puerta int. marco chapé hoja P/B 80x210	U	1	3850.00	Aberfuradasequipamientos
Generico	Portón garaje 3 hojas cristal 240x210	U	1	18200.00	Aberfuradasequipamientos
Generico	Placar integrar a alt. ancho 110 alt. 2.05	U	1	11658.00	Aberfuradasequipamientos
Generico	Placar integrar a alt. ancho 165 alt. 2.05	U	1	16070.18	Aberfuradasequipamientos
Generico	Placar integrar a alt. ancho 2.20 alt. 2.05	U	1	17433.23	Aberfuradasequipamientos
Generico	Placar integrar a alt. ancho 165 alt. 2.40	U	1	16344.89	Aberfuradasequipamientos
Generico	Placar integrar a alt. ancho 2.20 alt. 2.40	U	1	20934.28	Aberfuradasequipamientos
Generico	Cajón con llave ancho 50 cm	U	1	2444.40	Aberfuradasequipamientos
Generico	Bandejitas cantidad 3 altura total 50 cm	U	1	4044.90	Aberfuradasequipamientos
Generico	Balaníncio 80x80	U	1	1547.00	Aberfuradasequipamientos
Generico	Ventana 140x110	U	1	2550.00	Aberfuradasequipamientos
Generico	Puerta cocina 80x205	U	1	3400.00	Aberfuradasequipamientos
Generico	Celosía cl/lejido galvanizado 40x100	U	1	2830.00	Aberfuradasequipamientos

Create

Nombre: Daniel

Apellidos: Hernandez

Cedula: 3252561

Sexo: M

Direccion: 18 De Julio 1204

Pais: Uruguay

Ciudad: Centro

Imagen:

Edad: 34

Email: danny@hotmail.com

Rut: 352332

Create

Log in.

Use a local account to log in.

There are no external authentication services configured for this application. Set up one for details or setting up this ASP.NET application to support logging in via external services.

Email:

Password:

Remember me?

Log in

[Register](#) [Log in](#)

[Forgot your password?](#)

Copyright © 2015 IGAFAL by Getty RM

Fig. 73 Screenshots iniciales de requisitos para interfaz externa

2.14. Diagramas

2.14.1. Evolución de las entidades del modelo

2.14.1.1. Entidades del sprint 1 al 3

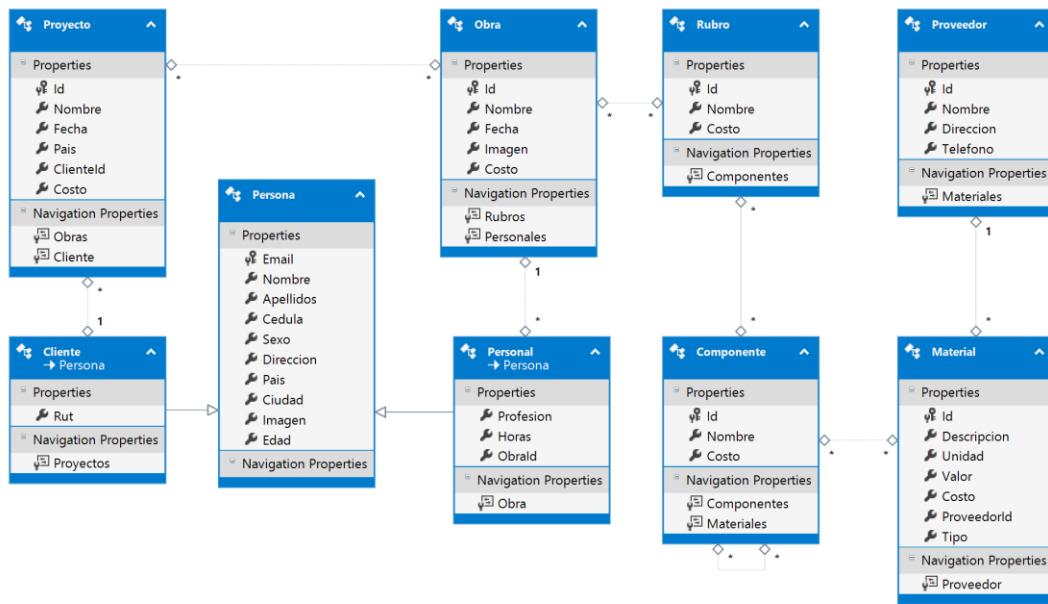


Fig. 74 Diagrama de evolución de las entidades del modelo del sprint 1 al 3

2.14.1.2. Entidades del sprint 3 al 4

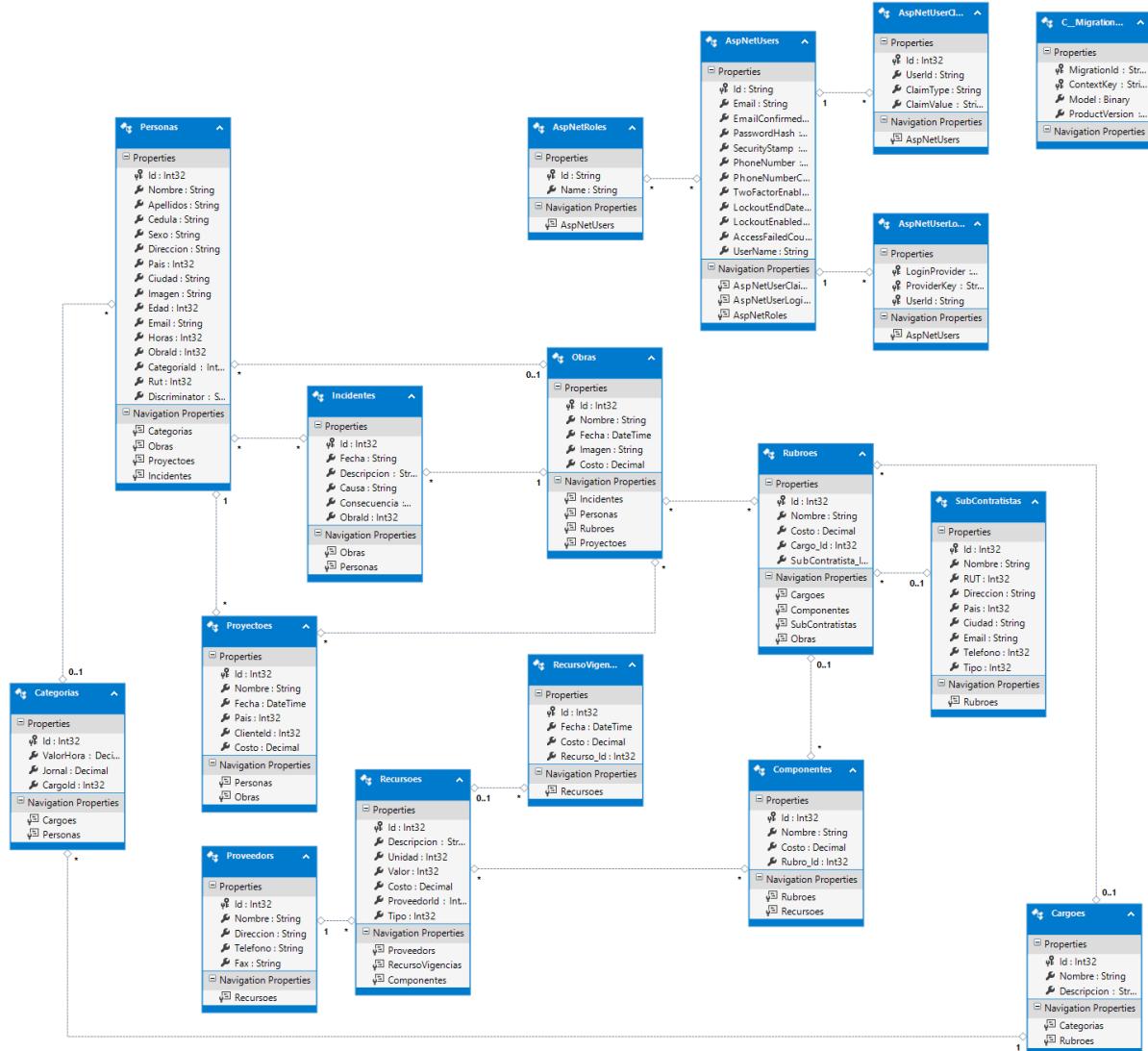


Fig. 75 Diagrama de evolución de las entidades del modelo del sprint 3 al 4

2.14.1.3. Entidades del sprint 4 al 6

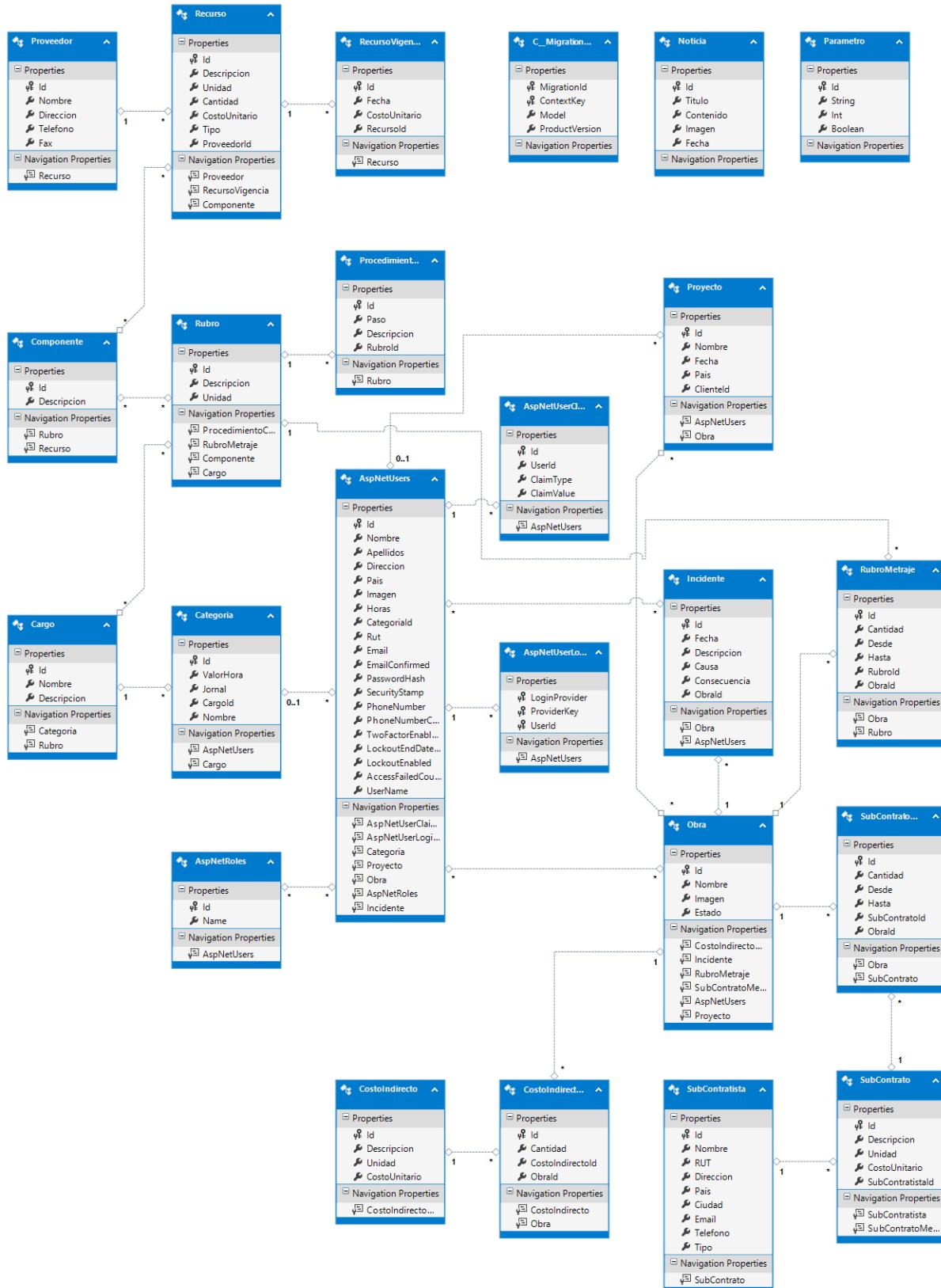


Fig. 76 Diagrama de evolución de las entidades del modelo del sprint 4 al 6

2.14.2. Secuencia

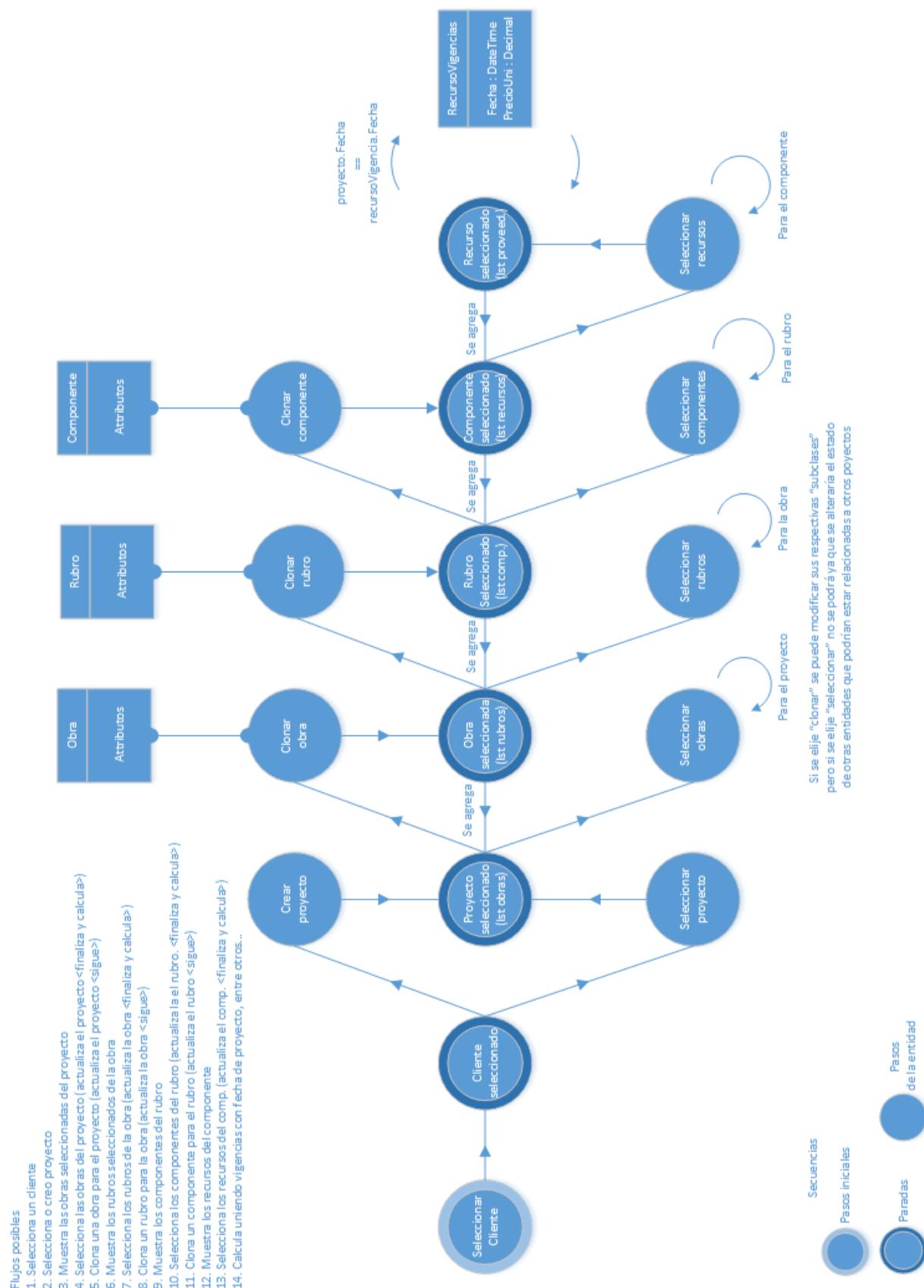


Fig. 77 Diagrama de secuencia para el funcionamiento de los wizards

2.14.3. Arquitectura

2.14.3.1. Diagrama de dependencias general

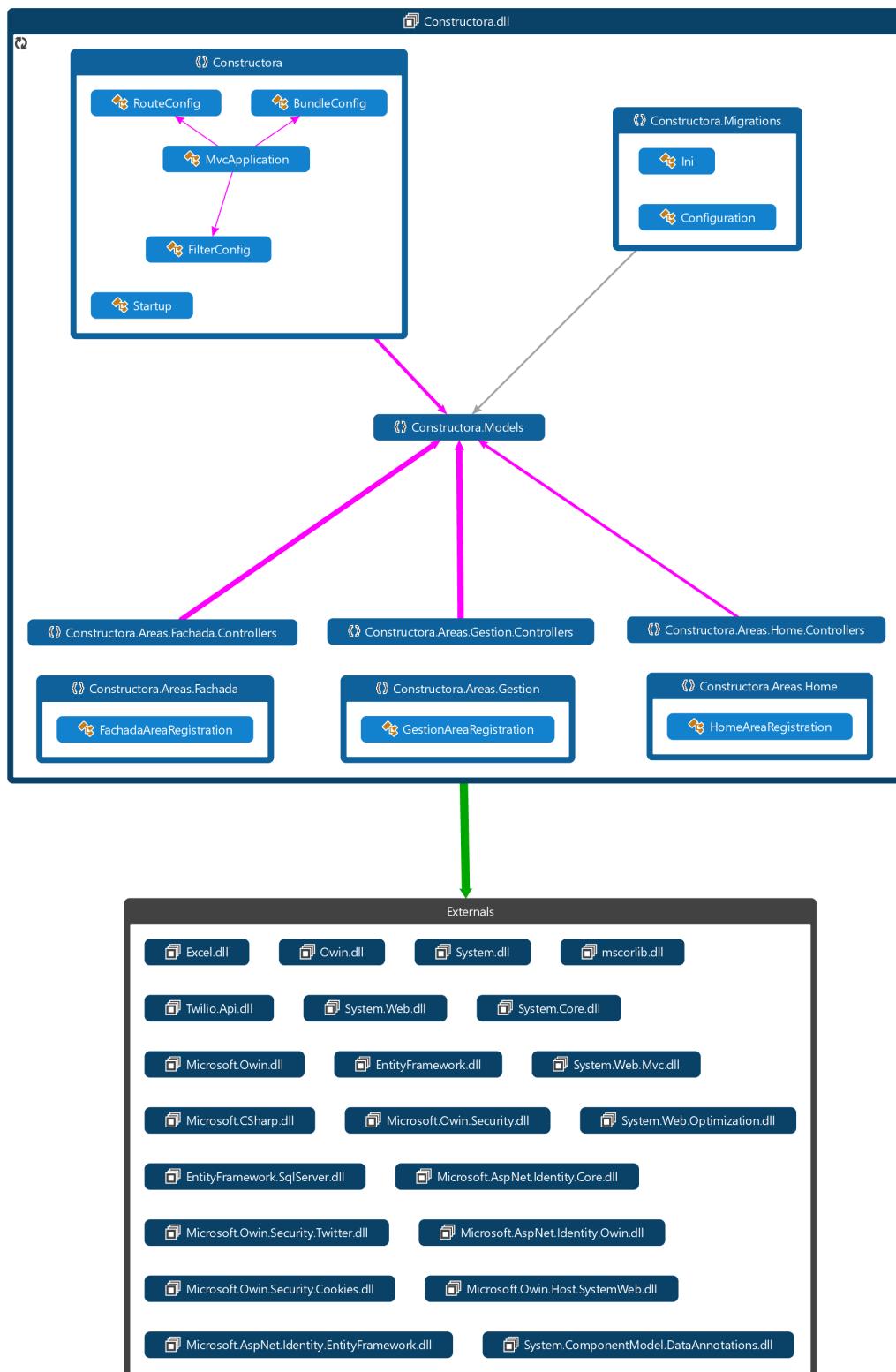


Fig. 78 Diagrama de dependencias general

2.14.3.2. Diagrama de dependencias con detalles en los controladores

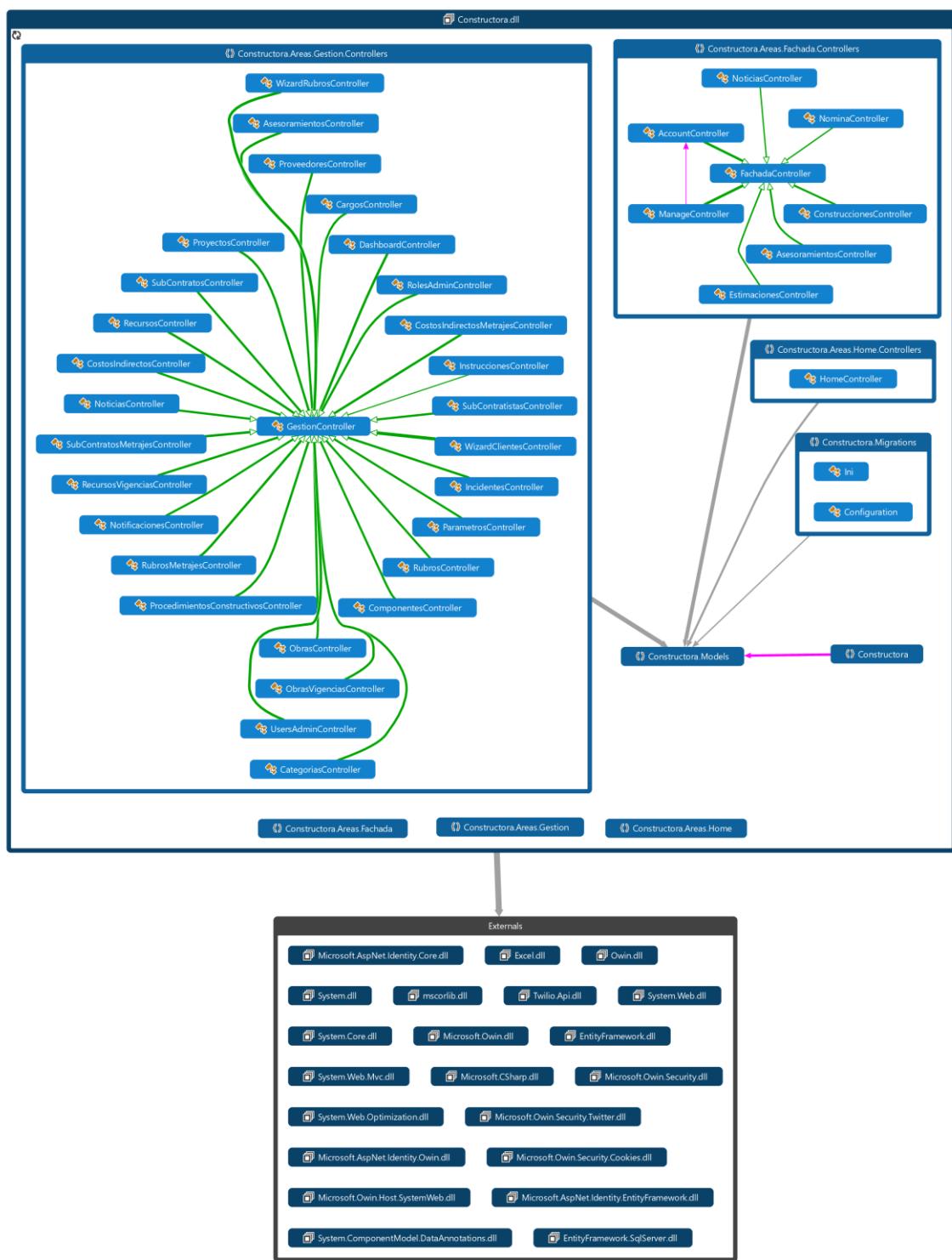


Fig. 79 Diagrama de dependencias con detalles en los controladores

2.14.3.3. Diagrama de dependencias con detalles en el modelo

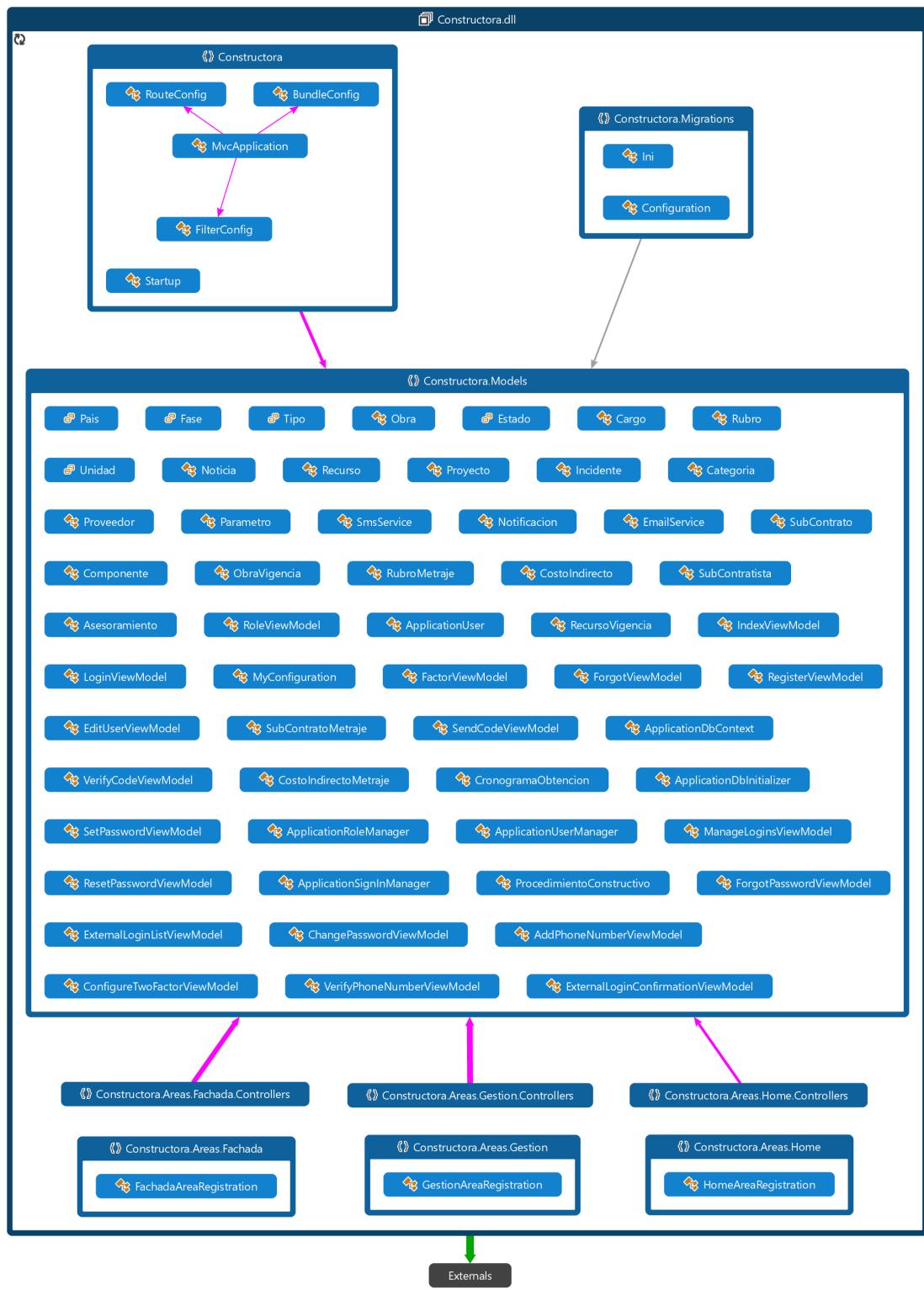


Fig. 80 Diagrama de dependencias con detalles en el modelo

2.15. Accesos web

El sitio se puede acceder mediante cualquiera de estas urls:

- <http://construccion.gabyrm.com/>
- <http://construccion.azurewebsites.net/>

La aplicación web se puede acceder con:

- <https://play.google.com/store/apps/details?id=com.gabyrm.construccion>

El usuario administrador que se usara para testear el sitio es:

- Usuario: info@gabyrm.com
- Contraseña: Admin01!

El FTP con el código fuente está alojado en:

- Host: <ftp://waws-prod-cq1-001.ftp.azurewebsites.windows.net>
- Usuario: Construccion\gabyrm
- Contraseña: UniversidadORT!

La db se puede acceder mediante SQL manager con:

- Host: construccion.database.windows.net
- Usuario: gabyrm
- Contraseña: UniversidadORT!

2.16. Despliegue para cada producto entregable

Para realizar un *deploy* del producto entregado se recomienda realizar los siguientes pasos.

Localizar una versión de la app dentro del historial.

Name	Date modified	Type	Size	Availability
201505152011.rar	5/15/2015 20:11	WinRAR archive	31,468 KB	Available offline
201505150126.rar	5/15/2015 01:26	WinRAR archive	31,464 KB	Available offline
201505150052.rar	5/15/2015 00:52	WinRAR archive	31,464 KB	Available offline
201505142012.rar	5/14/2015 20:12	WinRAR archive	31,000 KB	Available offline
201505130052.rar	5/13/2015 00:52	WinRAR archive	31,024 KB	Available offline
201505130042.rar	5/13/2015 00:42	WinRAR archive	31,025 KB	Available offline
201505130003.rar	5/13/2015 00:03	WinRAR archive	30,786 KB	Available offline
201505122010.rar	5/12/2015 20:09	WinRAR archive	30,560 KB	Available offline
201505112043.rar	5/11/2015 20:43	WinRAR archive	30,529 KB	Available offline
201505111945.rar	5/11/2015 19:45	WinRAR archive	30,526 KB	Available offline
201505111759.rar	5/11/2015 17:59	WinRAR archive	30,159 KB	Available offline
201505101926.rar	5/10/2015 19:26	WinRAR archive	30,159 KB	Available offline
201505101818.rar	5/10/2015 18:18	WinRAR archive	30,095 KB	Available offline

Fig. 81 Screenshot de historial de versiones

Una vez localizada una versión descomprimir y abrir el “.sln” dentro de la carpeta, se recomienda utilizar Visual Studio 2012 o 2013 obligatoriamente con todos los *package* y las actualizaciones de Azure.

.nuget	5/3/2015 12:18	File folder
Constructora	5/13/2015 00:40	File folder
packages	5/3/2015 12:18	File folder
Constructora.sln	5/3/2015 12:20	Microsoft Visual Stu... 2 KB

Fig. 82 Screenshot de la carpeta raíz de la solución

Una vez abierto el proyecto ir al “*solution explorer*” y sobre el nodo con el nombre del proyecto dar *click derecho* seleccionando “*publish*”.

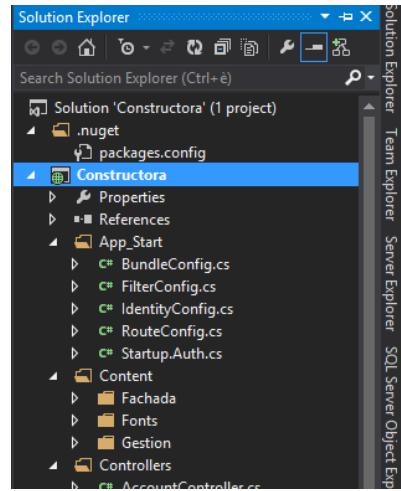


Fig. 83 Screenshot del solution explorer

Al entrar al “*publish*” *loguearse* con su cuenta de Azure (saldrá un *prompt* si no se está *logueado*) y seleccionar un *website* o una *virtual machine*.

Al hacerlo le ira pidiendo algunos datos sobre la instancia y tipo de suscripción que son totalmente independientes del proyecto.

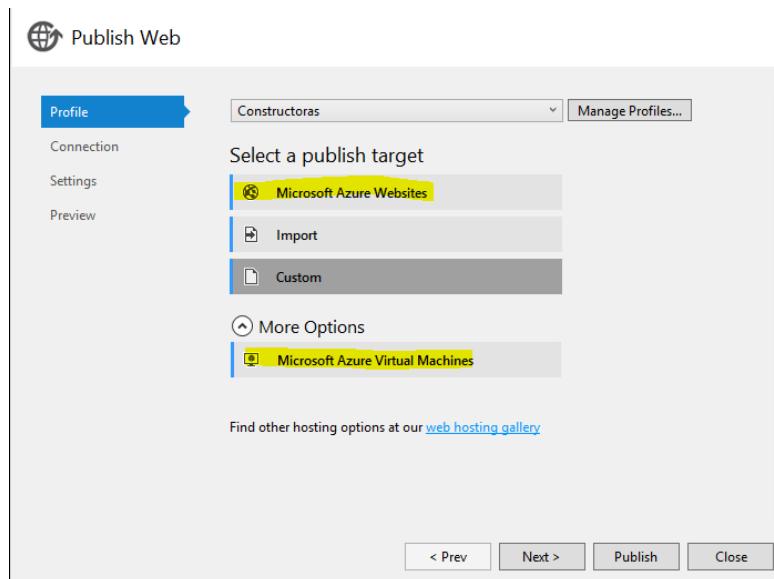


Fig. 84 Screenshot de publish web

Verificar que la conexión funcione yendo a la pestaña “connection” y clickeando sobre “validate connection”, esto es automáticamente configurado al realizar el paso anterior. De lo contrario se tendrán que ingresar manualmente.

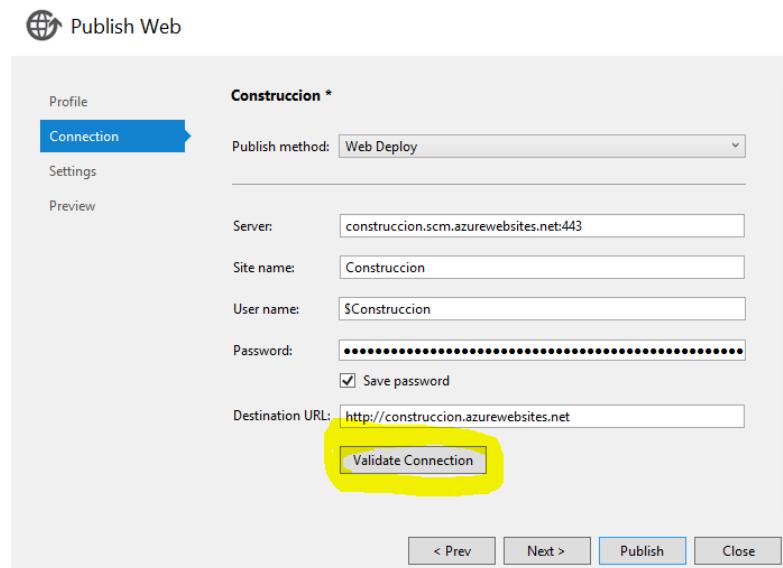


Fig. 85 Screenshot connection

Se está utilizando un web *deploy* con este manejador ya que es más seguro y rápido que un FTP. Observar que el número de puerto para la conexión es 443 esto está relacionado a que se utiliza el protocolo HTTPS “seguro” mediante SSL cifrando los datos para evitar posibles amenazas, evitando tomar riesgos innecesarios.

Una vez verificado lo anterior se procede a establecer la conexión con la base de datos, para esto se necesita crear un “connection string” que referencia a la dbContext que se utiliza en el proyecto. Puede ingresarla a mano o clickeando al lado del textbox le guiará por un proceso sencillo para conectarse con la db y generar dicho *string*. Para las claves de acceso a la db lea la sección anterior denominada “Accesos web”.

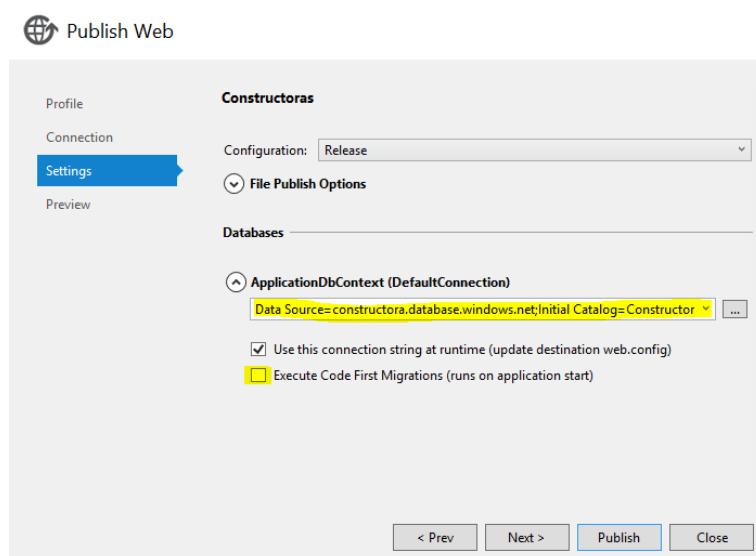


Fig. 86 Screenshot de settings

Lo siguiente es verificar que archivos se subirán y cuales no ya que son idénticos a los que existen en el servidor.

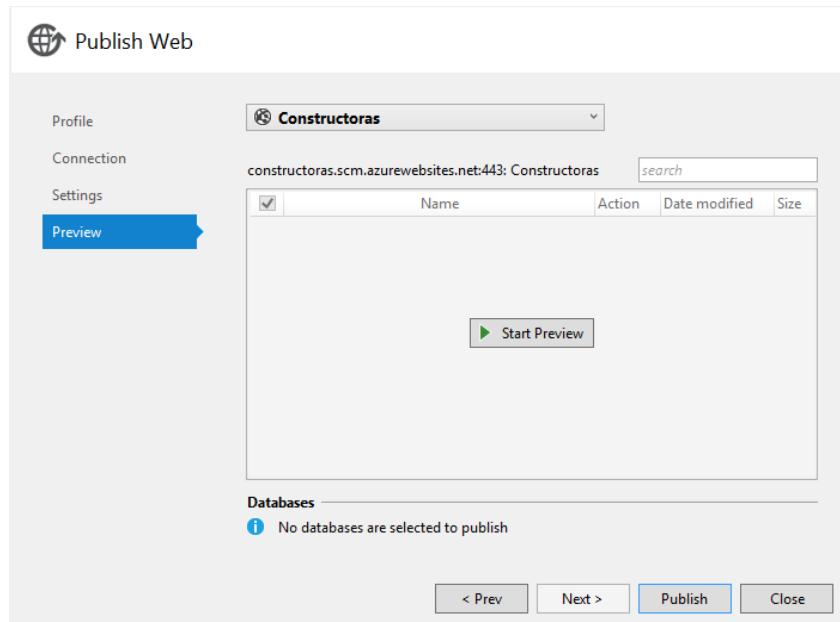


Fig. 87 Screenshot de preview

Por ejemplo, en este *deploy* se están subiendo “*add*” y actualizando “*update*” algunos archivos y dll compilados correspondientes a esta versión del proyecto.

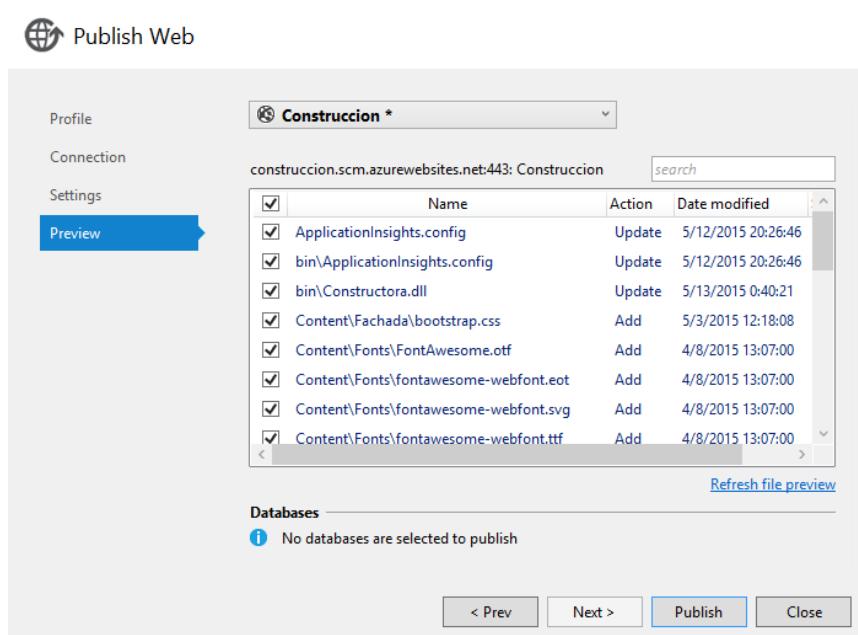


Fig. 88 Screenshot de preview verificando

Ahora tenemos nuestra app ya en la web, pero falta insertar en la base de datos las entidades correspondientes al modelo de Entity Framework.

Para esto ir al “*server explorer*” y agregar si corresponde la conexión a la base de datos de Azure.

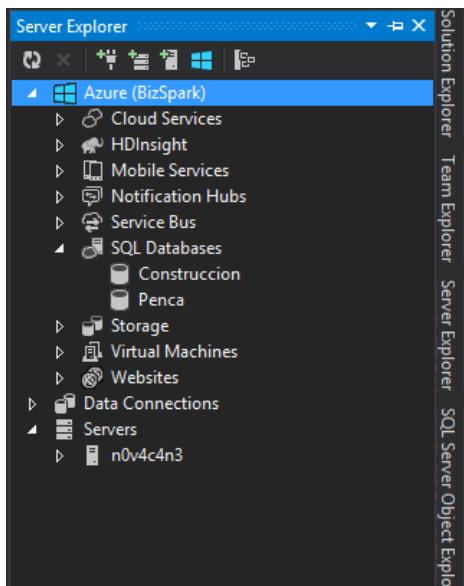


Fig. 89 Screenshot de server explorer

Alternativamente se podría utilizar la opción situada en el menú superior “*connect to database*” ingresando los datos de la cuenta de la base de datos descriptos en la sección anterior.

Una vez conectado con la base de datos, procederemos a impactar las “*migrations*” en la nueva base de datos, para ello abrir el “*package manager console*” e ingresar “*Update-Database*”, esto lo que hará será recorrer todas las clases de la carpeta “*migrations*” y verificar si fueron aplicadas o no a la base de datos (según si existe en la tabla historial de migraciones de la db), en nuestro caso no existirá y se procederá a ingresar desde la migración “*initial*” hasta la última, logrando así generar la estructura más reciente de la db que se tenía antes de archivar el proyecto.

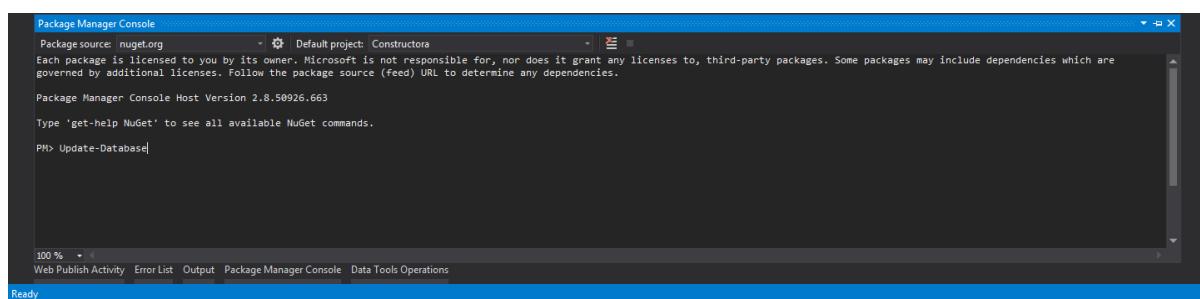


Fig. 90 Screenshot del output

2.17. Utilización de la nube de Azure

Para el desarrollo del sistema se utilizó Azure que cuenta con una interfaz web denominada Portal de administración que permite a los administradores tener acceso a la mayoría de características de Azure y administrarlas. Microsoft normalmente publica el portal de la IU más reciente en beta antes de retirar uno antiguo. La más reciente se llama "Portal de vista previa de Azure".

Se accede al portal mediante la siguiente pantalla.



Fig. 91 Screenshot del login Azure

Normalmente hay un largo periodo de solapamiento en el que los dos portales están activos. Aunque los servicios fundamentales aparecerán en los dos portales, es posible que no toda la funcionalidad esté disponible en ambos. Puede que los servicios más recientes aparezcan primero en el portal más nuevo y que la funcionalidad y los servicios más antiguos existan solo en el anterior. Lo que tiene que tener en cuenta es que, si no encuentra algo en el portal más antiguo, debe comprobar el más reciente (y viceversa).

En éste caso se mostrara el nuevo portal con el que se desarrolló la aplicación.

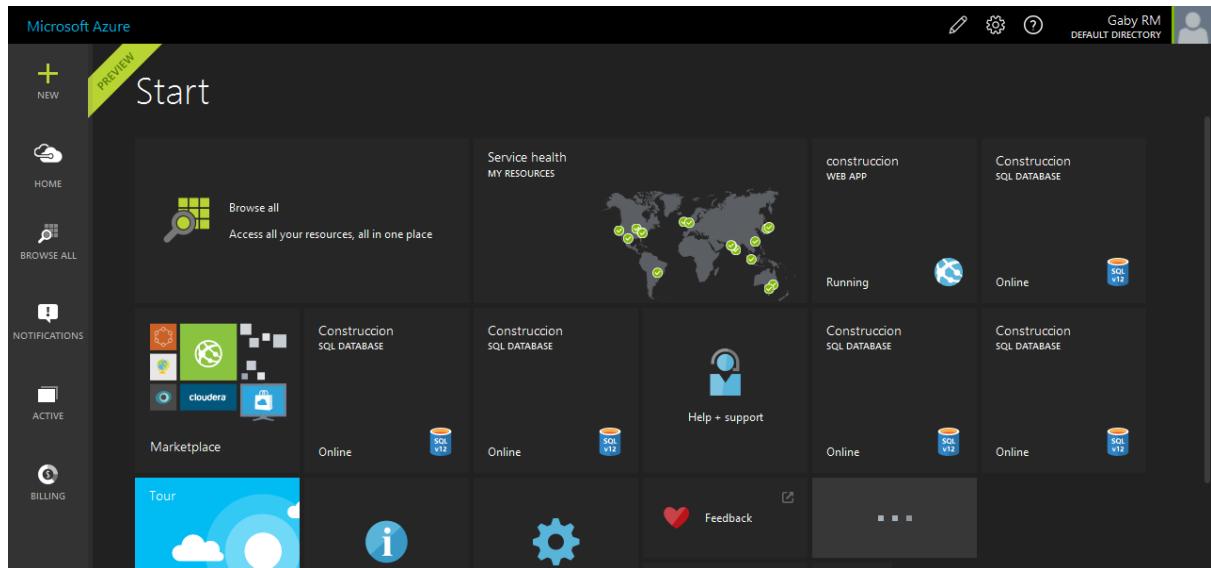


Fig. 92 Screenshot del dashboard azure

En el Portal de vista previa de Azure se puede crear grupos de recursos para administrar juntos todos los recursos de una aplicación. El grupo de recursos es un nuevo concepto en Azure que sirve como límite del ciclo de vida para cada recurso contenido en él.

Los grupos de recursos le permiten administrar juntos todos los recursos en una aplicación. La nueva funcionalidad de administración, Resource Manager de Azure, permite el uso de grupos de recursos. Resource Manager de Azure permite agrupar varios recursos como grupo lógico que sirve como límite del ciclo de vida para cada recurso contenido en él. Normalmente, un grupo contendrá recursos relacionados con una aplicación específica. Por ejemplo, un grupo puede contener un recurso de sitio web que hospeda el sitio web público, una Base de datos SQL que almacena datos relacionales que usa el sitio y una cuenta de almacenamiento que conserva recursos no relacionales.

Para el desarrollo de este proyecto se creó y utilizó el siguiente grupo.

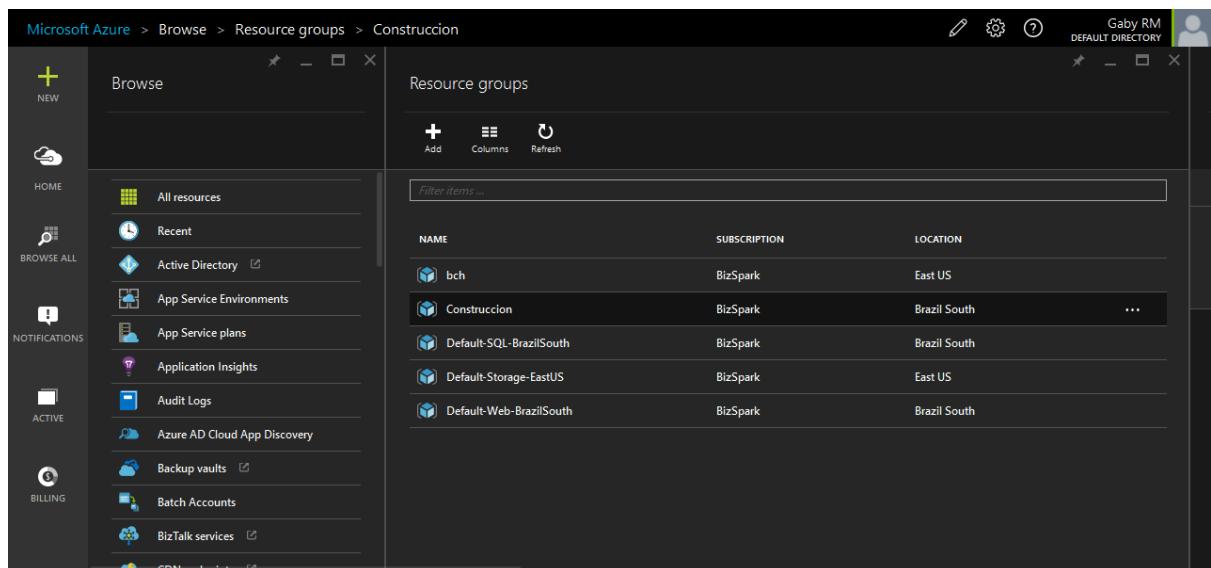


Fig. 93 Screenshot de resource groups

Una vez seleccionado el grupo se puede observar la sección de resumen muestra una asignación visual de todos los recursos del grupo en cuestión. También muestra los recursos de otros grupos que están vinculados a este grupo de recursos. La asignación de recursos también muestra el estado de cada uno de ellos.

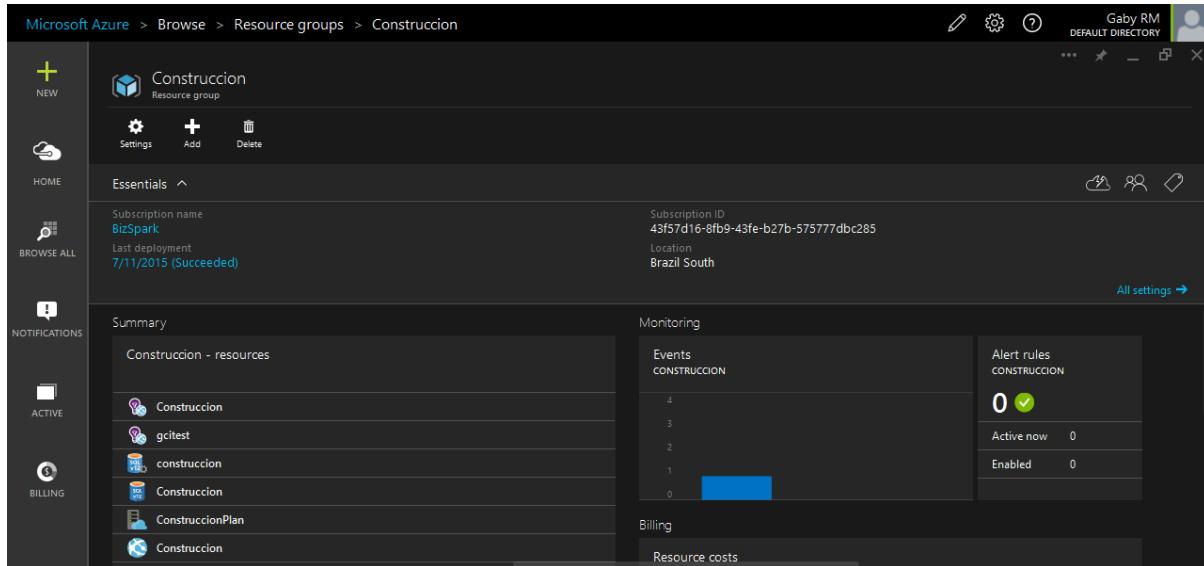


Fig. 94 Screenshot de resource groups seleccionado

Una de las cosas más importantes que se hacen en la nube es ejecutar sitios y aplicaciones web. Máquinas virtuales de Azure le permite hacer esto, pero le deja la responsabilidad de administrar una o varias VM y los sistemas operativos subyacentes. Los roles web de servicios en la nube pueden hacerlo, pero su implementación y mantenimiento conlleva también trabajo administrativo.

Esto es exactamente lo que proporciona Aplicaciones web. Este modelo de proceso ofrece un entorno web administrado utilizando el Portal de administración de Azure y las API. Puede mover una aplicación de sitio web existente a Aplicaciones web de Azure sin cambios, o puede crear una directamente en la nube. Cuando un sitio web está en ejecución, puede agregar o quitar instancias de forma dinámica y recurrir a Aplicaciones web de Azure para equilibrar las solicitudes de carga entre ellas. Aplicaciones web de Azure ofrece tanto una opción compartida, en la que el sitio web se ejecuta en una máquina virtual con otros sitios, como una opción estándar, que permite que un sitio se ejecute en su propia VM. La opción estándar también le permite aumentar el tamaño (potencia informática) de las instancias, si es necesario.

Para el desarrollo, Aplicaciones web es compatible con .NET, PHP, Node.js, Java y Python, junto con Base de datos SQL y MySQL para el almacenamiento relacional. También ofrece compatibilidad integrada con varias aplicaciones conocidas, como WordPress, Joomla y Drupal. El objetivo es proporcionar una plataforma de bajo coste, escalable y enormemente útil para la creación de sitios y aplicaciones web en la nube pública.

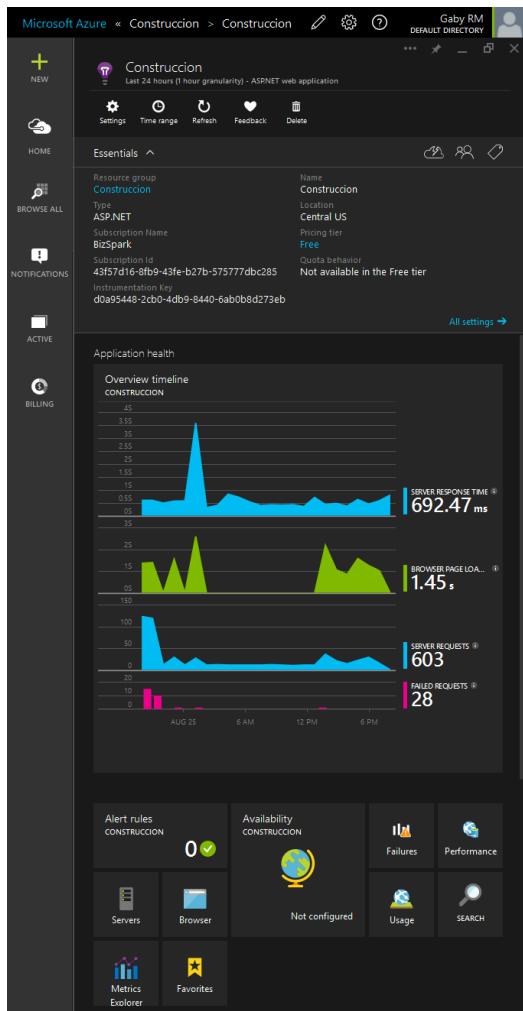


Fig. 95 Screenshot de application inisghts del proyecto

Application insights del proyecto

Application Insights de Visual Studio fue utilizado para supervisar la aplicación activa para ayudar a detectar y diagnosticar problemas y excepciones de rendimiento, y descubrir cómo se usa la aplicación. Se puede usar con una amplia variedad de tipos de aplicación. Funciona con las aplicaciones hospedadas en máquinas virtuales de Azure o en servidores IIS locales propios, así como con las aplicaciones web de Azure.

Para hospedar la estructura del proyecto se utilizó Aplicaciones web de Azure, las cuales permiten crear esta clase de aplicación web.

Esto es lo que proporciona servicios en la nube de Azure. Esta tecnología está diseñada expresamente para permitir aplicaciones escalables, confiables y de bajo coste administrativo, y es un ejemplo de lo que comúnmente se denomina plataforma como servicio (PaaS). El código se ejecuta en máquinas virtuales (conocidas como instancias) que ejecutan una versión de Windows Server.

Pero estas VM son distintas de aquellas que se crean con Máquinas virtuales de Azure. Y la diferencia es que Azure es quien las administra. Así, realiza tareas como instalar revisiones de sistemas operativos e implementar automáticamente las nuevas imágenes revisadas. Esto implica que su aplicación no debe mantener el estado en las instancias de rol web o de trabajo, sino que, por el contrario, se mantiene en una de las opciones de administración de datos de Azure

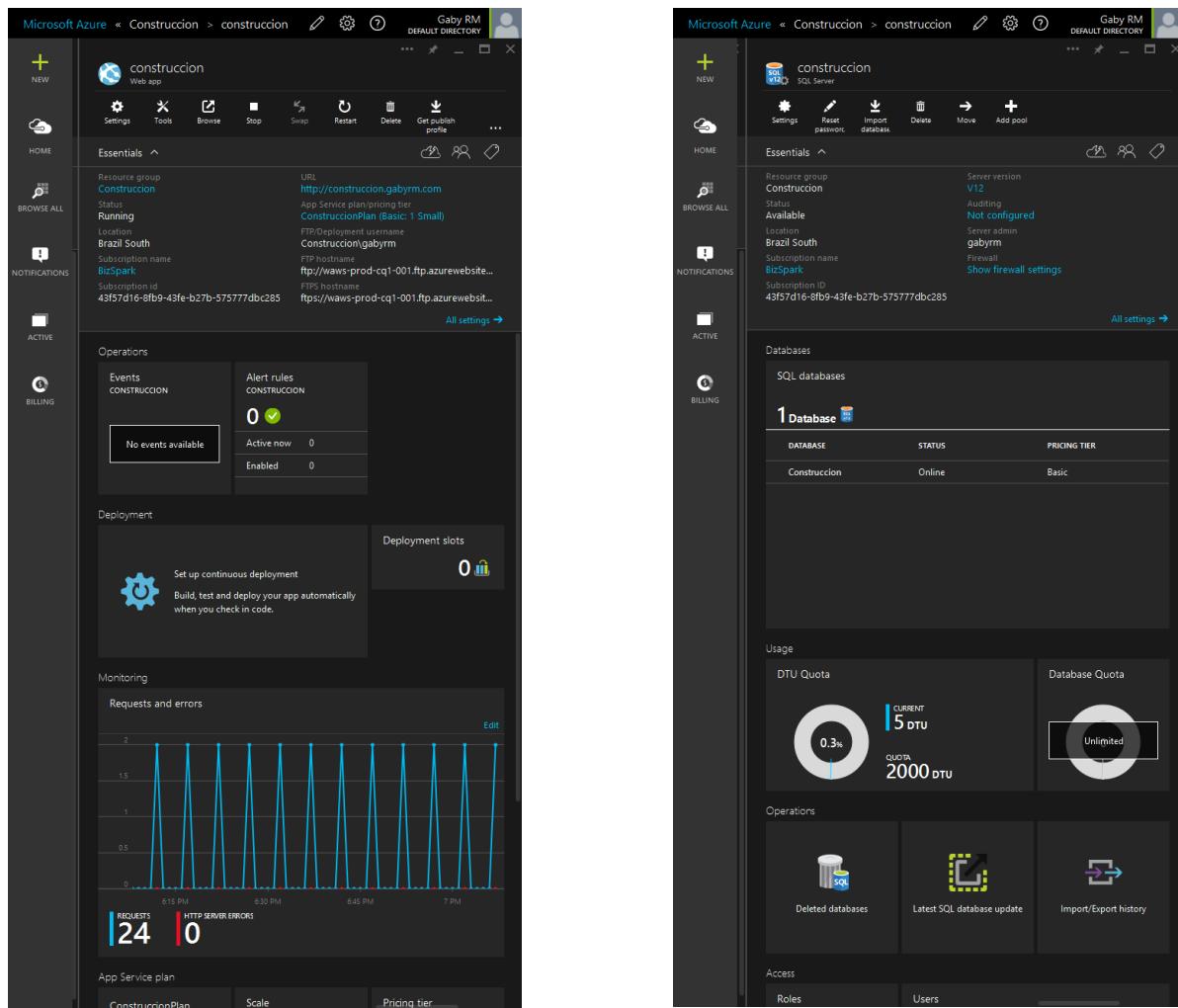


Fig. 96 Screenshot de monitoreo de servicios

2.18. Aplicación web

A continuación, se dará un breve recorrido sobre algunas de las funcionalidades principales.

Página de login

The screenshot shows the login page for IGAFAL S.A. at [http://igafal.com.ar](#). The top navigation bar includes links for 'Noticias', ' Nómina', 'Sobre nosotros', 'Registro', and 'Log in'. The main content area is titled 'Log in' and contains two sections: 'Usa una cuenta local para loguearte' and 'Usa otro servicio para loguearte'. The local login section has fields for 'Email' and 'Password', a 'Remember me?' checkbox, and an 'Ingresar' button. Below these are links for '¿Registrarse como nuevo usuario?' and '¿Olvidaste tu contraseña?'. The external login section features a 'Twitter' logo with the text 'Twitter' next to it, accompanied by explanatory text about using Twitter for login. At the bottom, there is a copyright notice: 'Copyright © 2015 IGAFAL S A by Gaby RM'.

Fig. 97 Página de login

En esta página se puede acceder mediante cualquier par de email/contraseña al sistema. Dependiendo del rol del usuario se mostrarán u ocultarán acciones, las mismas no solo serán ocultadas visualmente, sino que, si el mismo intenta acceder mediante URL a alguna acción que no posea permisos, se le redireccionará al *login* con un “*returnurl*”, la cual si logra acceder con una cuenta que posea los permisos, se lo dirigirá a la “*returnurl*” y si no volverá a la página de bienvenida por defecto.

El usuario puede acceder al sistema utilizando una cuenta externa, en este caso se habilitó el acceso utilizando Twitter, si el usuario posee una cuenta en Twitter la puede vincular o registrarse al sitio utilizándola. Para lograr esto se creó una app en Twitter Dev Apps cuyo nombre es “Constructora” y tiene habilitado el *login* externo, un *api key* y un *secret key* que se usa en conjunto con el componente OWIN.

OWIN es una especificación abierta, que pretende reducir la dependencia de las aplicaciones del servidor web o de aplicaciones. Un poco como ya hacen node.js o Rack. Antes de OWIN las aplicaciones web estaban montadas sobre una pila parecida a “Aplicación” -> “ASP.NET” -> “IIS”, Esto implicaba un sistema demasiado estricto y muy difícil de evolucionar. OWIN trata de evitar estos problemas, con una aproximación diferente.

Los componentes que ahora tenemos son los siguientes:

- Host: sería la capa más cercana al sistema operativo, encargada de gestionar los procesos que ejecutarán las aplicaciones.
- Server: componente utilizado para escuchar las peticiones HTTP.
- Middleware: componente intermedio que se encarga de procesar las peticiones para realizar alguna operación sobre ellas. Aquí entrarían *frameworks* y componentes como SignalR, Web API, Authentication etc.
- Aplicación: la aplicación desarrollada que hace uso de los otros elementos.

Lo bueno de estar arquitectura, es que, al estar desacoplada, siempre podemos sustituir cualquier componente, por otro que haga la misma función. Por ejemplo, como host podemos usar IIS (que también puede hacer de Server), pero también podemos usar OwinHost, que es un simple ejecutable que puede servir de host a las aplicaciones. La idea es que con esta arquitectura podamos conseguir que la aplicación funcione en cualquier plataforma con mucha más facilidad.

Si el usuario no opta por utilizar el *login* externo su contraseña será encriptada y almacenada en la base de datos ubicada en Azure.

Página de registro

The screenshot shows a registration form titled "Registro". At the top, there is a navigation bar with links for "IGAFAL S.A", "Noticias", " Nómina", "Sobre nosotros", "Registro", and "Log in". The main form area has a header "Completa los siguientes datos para registrarte como cliente de nuestro sistema." Below this, there are input fields for "Nombre" (Gabriel), "Apellidos" (Martinez Morganti), "Direccion" (26 De Marzo 1297/202), "Pais" (Argentina), "Imagen" (a placeholder link), "Rut" (1591239346), "Email" (info@gabyrm.com), "Password" (*****), and "Confirm password" (*****). A "Registrar" button is at the bottom right. At the very bottom of the page, there is a small copyright notice: "Copyright © 2015 IGAFAL S.A by Gaby RM".

Fig. 98 Screenshot de página de registro

En esta página se puede registrar cualquier usuario, el mismo después de completar los datos como; nombre, apellidos, dirección, país, imagen (como link), rut, email, contraseña y confirmación de contraseña se podrá registrar como usuario del tipo “cliente” del sistema. Esto les brindara acceso a las funciones públicas para todo cliente, ya sea pedir asesoramientos, ver estimaciones, ver planes de construcción, incidentes, gestionar sus proyectos entre otros.

Inicialmente el usuario se registra con rol de cliente, pero esto se puede modificar posteriormente en el sistema de gestión integrada quitando o asignando roles según corresponda.

Al registrarse le aparecerá una página de registró exitoso y se le enviará un email para confirmar su cuenta, siguiendo el link del mail se le mostrará una página diciendo que se ha confirmado su cuenta y se lo invitara a iniciar sesión.

Página de perfil y tareas

IGAFAL S.A. Noticias Nómina Sobre nosotros Perfil y tareas Gestión Log out

Mi perfil

Gabriel Martinez
Uruguay

Dirección 26 De Marzo 1297/202
Rut 11241251250
Email info@gabym.com
EmailConfirmed
AccessFailedCo... 0

Configuración de perfil
Donde se podrá configurar todos los aspectos relacionados con tu cuenta.

Mis tareas

ESTIMACIONES
Para ver información sobre todos las consultas de asesoramiento en proyectos que fueron asignados

CONSULTAS
Para enviar una consulta de asesoramiento para una obra en particular que quieras llevar a cabo en

CONSTRUCCIONES
Para ver el estado de todos los proyectos en desarrollo que has concretado con IGAFAL S.A. ya

Fig. 99 Screenshot de página de perfil y tareas

En el perfil se engloba las principales funcionalidades que puede acceder un cliente o un usuario de rango superior. Se muestra, por un lado, la imagen de perfil, el nombre y país. Por otro lado, los datos generales del usuario, un link a configurar el perfil, donde podrá cambiar varios aspectos del comportamiento de su cuenta, como la verificación en dos pasos, recordar el browser, agregar número de teléfono celular, cambiar contraseña, etc.

En la parte inferior se presenta la sección de “Mis tareas” donde se agrupa de manera simple y funcional las principales funcionalidades del sistema como:

Las estimaciones, para ver información sobre todas las consultas de asesoramiento en proyectos que fueron asignados a la cuenta del usuario.

Las consultas, para enviar una consulta de asesoramiento para una obra en particular que se quiera llevar a cabo en IGAFAL S.A. y por ultimo las construcciones, para ver el estado de todos los proyectos en desarrollo que se han concretado con IGAFAL S.A, ya sea por consulta u otro medio.

Páginas de asesoramientos

Fig. 100 Screenshots de página de registro

Informar al usuario y la auto-documentación del proyecto fue un punto primordial al momento de desarrollar todas las funcionalidades del sistema, para ello se creó un estándar para la ubicación de todos los mensajes informativos que guíen al usuario en cada pantalla. Éstos están generalmente a la derecha del formulario, encima de una grilla o como instructivos de pantalla completa si la funcionalidad es de prioridad alta.

En este caso para los asesoramientos se creó un instructivo de pantalla completa dividido en secciones tales como “Qué”, “Cómo”, “Cuáles” y “Cuándo” que intenta explicar lo más clara y concisamente todo el proceso de estimación de costos visto desde la perspectiva del cliente.

Si el cliente continúa presionando sobre el botón de “+” se lo guiará a los pasos para crear una consulta de asesoramiento, donde el mismo especificará todos los datos referentes a su proyecto. Una vez completados todos los datos se les enviará un mail a todos los usuarios administradores, se creará una notificación que saldrá en el *dashboard* y será la primera notificación en salir del *dropdown* del *layout* del sistema de gestión integrada.

El administrador deberá seleccionar el cliente y asignarle obras al proyecto inicializado al momento de pedir el asesoramiento. Una vez realizado todo el proceso de estimación, el administrador puede notificar al cliente que la estimación para su proyecto fue calculada y enviarle un link para ir directamente al proyecto.

Este link es accesible tanto si el usuario al momento de hacer *click* tiene una sesión activa o si está sin una sesión, ya que como se explicó en la sección del *login*, utiliza un “*returnurl*” el cual lo redireccionará a la pantalla de *login* si no posee sesión activa y redireccionará al proyecto cuando la posea.

Páginas de estimaciones del cliente

IGAFAL S.A. Noticias Nómica Sobre nosotros Perfil y tareas Gestión Log out

Estimaciones

Estos son todos los proyectos que realizaste una consulta de asesoramiento o que fueron asignados directamente por nuestro sistema. Selecciona uno para ver su estimación.

Nombre	Fecha	País	Fase
Parrillero de casa	8/4/2015 9:10:19 PM	Uruguay	Estimando Estimación
Galana	1/8/2015 12:00:00 AM	Argentina	Construyendo Estimación
test mail	8/14/2015 1:38:06 AM	Argentina	Construido Estimación
casita	8/25/2015 2:20:01 AM	Argentina	Estimando Estimación

Copyright © 2015 IGAFAL S.A by Gaby RM

IGAFAL S.A. Noticias Nómica Sobre nosotros Perfil y tareas Gestión Log out

Casa Galana \$ 40135.95

Detalles

Cocina marmol Cantidad: 10.00 M2 Desde: 6/4/2015 12:00:00 AM Hasta: 6/4/2015 12:00:00 AM Piso baldosa porcelanato Apliación de montero monocapa liso o raspado Cantidad: 1.00 M2 Proveedor: Genérico Costo unitario: \$ 1015.03 PORTLAND BLANCO Cantidad: 1.00 KG

miku kano Celular: Email: miku@getgym.com Dirección: 18 De Julio 12044405 Jornal: \$

Luz Cantidad: 300.00 HORAS Costo unitario: \$ 300.00

Ranuramiento

IGAFAL S.A. Noticias Nómica Sobre nosotros Perfil y tareas Gestión Log out

Tu consulta fue enviada Te notificaremos por email... Estimaciones

Copyright © 2015 IGAFAL S.A by Gaby RM

Fig. 101 Screenshots de páginas de estimaciones del cliente

En estas 3 imágenes se puede observar cómo funciona la visualización del proceso de estimación de costos para proyectos del lado del cliente. Al realizar una consulta de asesoramiento se crea un proyecto en fase “estimando” y se lista dentro de las estimaciones del cliente.

Si el cliente selecciona para ver una estimación para un proyecto en fase “estimando”, visualizará una pantalla con una animación de “enviando” y una descripción avisando que se enviará un mail cuando se calcule una estimación para el proyecto.

Cuando el cliente selecciona un proyecto que esté en fase “estimado”, “construyendo” o “construido” podrá visualizar la estimación calculada agrupada en tabs con el nombre de cada obra que se asignó al proyecto.

Dentro de cada tab se puede visualizar la estimación total, el nombre de la obra y un listado de los metrajes realizados a los rubros, con fechas de realización, un listado de los componentes del rubro y dentro de los componentes están los recursos que conforman al componente.

Por otro lado, se puede visualizar los empleados asignados a la obra, los costos indirectos y subcontratos necesarios para el desarrollo de la obra. Cada uno de estos “ítems” se representa con un ícono diferente.

Página de dashboard y layout del sistema de gestión integrada



Fig. 102 Screenshots de página de dashboard y layout del sistema de gestión integrada

En estas imágenes se puede visualizar el *layout* que se utiliza en todo el modulo del sistema de gestión integrada. La capacidad del *layout* de Razor permite definir una plantilla común para el sitio, y heredar su *look&feel* por todas las vistas/páginas.

Razor permite definir múltiples secciones, no continuas en las plantillas de *layout* que pueden ser rellenas por las plantillas de vistas. La sintaxis @section{} para esto es limpia y concisa. Razor también tiene la habilidad de permitir comprobar en tiempo de ejecución si una sección particular ha sido definida, y ofrecer un contenido alternativo (o incluso un diseño diferente) en caso de que no se indique. Esto ofrece una forma poderosa y fácil de personalizar el aspecto de nuestro sitio y lo hace más limpio y DRY desde el punto de vista de la implementación.

En el primer *dropdown* se puede visualizar las acciones “personales” del usuario como, configurar su cuenta, gestionar usuarios y roles, gestionar roles, configurar parámetros del sistema, ver incidentes y *desloguearse*.

En el segundo *dropdown* se muestran las 5 últimas notificaciones del sistema, las cuales sirven para mantener informado al administrador sobre todo lo que sucede en el sistema, se puede *clikear* sobre alguna para ver detalles o seleccionar ver todas para listar y trabajar con todas las notificaciones del sistema.

En el tercero *dropdown* se listan los 3 últimos incidentes para las obras del sistema, se ubicaron por separado ya que tiene prioridad las incidencias y todas las inquietudes que el cliente quiera reportar sobre las obras.

Páginas de instrucciones para el proceso de estimación y desarrollo de obras

IGAFAL S.A.

Inicio

Dashboard

Módulos

Instrucciones

Principales

Wizard clientes

Wizard rubros

Complementarias

Gestión Empresarial

Gestión Material

Gestión Humana

Proceso de estimación y desarrollo de obras

El sistema de cálculo de costos y gestión de obras permite determinar el precio total detallado de una obra, el consumo de materiales y mano de obra por categorías. Para sintetizar este hecho, prácticamente se emplea un formato denominado estimación de obra, el cual es el documento que nos permite llevar el control financiero de una obra y en el cual se resumen los conceptos, cantidades de obra ejecutada y sus importes para un periodo de tiempo determinado, así mismo, es el documento que permite al contratista presentar su requerimiento de cobro al Contratante con objeto de contar con la solvencia necesaria y no afectar el programa general de ejecución de la obra. Para ello se partirá de la siguiente información.

Paso 1 Recursos y proveedores
Aquí se ingresa los materiales y tipos de mano de obra que componen los componentes del siguiente paso. El Operador sólo deberá ingresar o actualizar, los costos unitarios de cada material o el costo horario de la mano de obra (excl.IVA) en sus diversas categorías asignándolos a sus respectivos proveedores.
[Proveedores](#) [Recursos](#)

Paso 2 Personalizar rubros
Aqui se inicializa los rubros que normalmente ejecuta en la empresa constructora, excluyendo los subcontratos. El operador deberá inicializar un rubro, armar sus componentes y seleccionar los recursos previamente creados en el paso 1. Estas entidades serán utilizadas para generar el "metraje" o el cómputo métrico de cada uno de los rubros de la obra.
[Rubros](#)

Paso 3 Empleados
Opcionalmente aquí se puede ingresar los empleados que ejecutarán las obras de los clientes del siguiente paso. Para esto se debe ingresar primero los cargos, después las categorías y por último asignar las categorías a un empleado.
[Cargos](#) [Categorías](#) [Usuarios](#)

Paso 4 Subcontratistas
Aqui se inicializan los subcontratistas los cuales son contratados por un contratista general o el contratista principal, para realizar una tarea específica como parte del proyecto global. Un subcontratista es asignado, ya sea para reducir costos o para mitigar los riesgos del proyecto. De esta manera, la empresa recibe el mismo servicio o mejor.
[Subcontratistas](#)

Paso 5 Estimación obras
Esta sección puede ser ampliada en todas y cada una de sus partes y modificada a voluntad por el operador para un cliente, agregando rubros y análisis, cambiando los metrajes, rendimientos para los rubros, de acuerdo a las informaciones, modalidades y criterios de cada uno.
[Clientes](#)

Fig. 103 Screenshot de páginas de instrucciones para el proceso de estimación y desarrollo de obras

Página de instrucciones para otras funcionalidades del sistema

Fig. 104 Screenshot de página de instrucciones para otras funcionalidades del sistema

Como se ha mencionado anteriormente uno de los puntos fuertes de este proyecto, es la auto documentación del sistema como una constante que ayude tanto al usuario del sistema en la utilización del mismo como a la persona encargada de desarrollarlo para mantener un registro claro y conciso de todas las funcionalidades del sistema.

Una de las opciones para realizar esto era crear una “wiki” para todo el sistema, pero la idea se fue mejorando hasta llegar al punto tal de crear un sistema que esté explicado en las mismas funcionalidades y que el punto de entrada para las diferentes funcionalidades sea la misma documentación, con esto se evita que el usuario tenga que leer largas, tediosas wikis y sepa en cada momento la descripción del lugar en el que está situado.

Por lo general dichas explicaciones o instrucciones están en las partes superiores o laterales del sistema ubicadas en *divs* grises.

Las instrucciones de un sistema son esenciales para proporcionar entendimiento de a quien lo vaya a usar para mantenerlo, para permitir auditorias del mismo y para enseñar a los usuarios como interactuar con él y a los operadores y administradores como hacerlo funcionar.

Tienen una gran importancia dentro de una empresa, ya que esta ayuda a eliminar la posible dependencia que se pueda formar entre el proyecto realizado, y el ejecutor de éste. Para que toda aplicación tecnológica y todo servicio informático queden adecuadamente documentado.

Páginas de wizard para clientes

The screenshots illustrate the 'Client - Wizard' interface for estimating costs. The first two screens show the selection of a client and the assignment of projects. The third screen provides a detailed view of a selected client's profile. The fourth screen shows a list of projects. The fifth screen shows a list of tasks or rubrics. The sixth screen shows a detailed view of a specific task.

Fig. 105 Screenshots de páginas de wizard para clientes

Estas son algunas de las páginas que componen al *wizard* para clientes, en él se concentra y encapsula todas las acciones relativas a la estimación de costos, se empieza por un listado de todos los usuarios del sistema a los cuales se les puede inicializar un proceso de estimación. Presionando en continuar sobre alguno de los usuarios se puede trabajar con los proyectos y asesoramientos de dicho usuario. Al seleccionar un proyecto ya sea por asesoramiento o por todos los proyectos del usuario, se podrá ver la página de asignación y creación de obras donde se podrá trabajar con dicho contenido, seleccionando las ya existentes, crear obras de cero, copiar obras, desasignar obras y filtrar o buscar las obras del cliente.

Una vez seleccionada una obra se llegará al paso de personalización de obra, el cual en una serie de *tabs* se podrá componer y personalizar todos los aspectos de la obra para generar una estimación y trabajar con todo lo que implique interactuar con el cliente de la obra, ya sean incidentes, procedimientos constructivos, asignación de metrajes de rubros a la obra, costos indirectos o subcontratos. Aparte se puede observar con una gráfica sencilla un estatus general de la obra y previsualizar la estimación total o generar el cronograma de obtención para un mes dado.

Cada una de estas pantallas se puede filtrar y ordenar por cualquier columna. El usuario puede ver el avance del proceso de estimación en la barra superior donde podrá observar que pasos completó y que pasos faltan completar para conseguir el resultado esperado.

Páginas de wizard para rubros

The screenshots illustrate the implementation of a wizard for managing categories (rubros). The first two screenshots show the 'Rubro - Seleccionar' (Rubro Selection) page, which allows users to create new categories and manage existing ones. It includes fields for description, unit, and continuation links. The third screenshot shows the 'Procedimiento constructivo' (Construction Procedure) page, where users can add steps to a procedure. The fourth screenshot shows the 'Rubro - Seleccionar' page again, this time with a focus on components.

Fig. 106 Screenshots de páginas de wizard para rubros

Para mantener una misma metodología de trabajo se implementó el *wizard* para rubros de manera muy similar al *wizard* para clientes. En el mismo se encapsulan todas las acciones referentes a los rubros. Donde se puede crear un rubro con una unidad, descripción, seleccionarlo para trabajar con su contenido, en el contenido del rubro, el cual está dividido en *tabs*, se puede agregar o quitar recursos y crear procedimientos constructivos.

De la misma manera que se indicaba los pasos que se deben seguir para completar el *wizard* para clientes, en el *wizard* para rubros se implementó la misma barra de estado con los pasos referentes al proceso de trabajo con los rubros.

Una vez creados los rubros se puede continuar con la creación de obras, si entre medio de la creación de una obra se necesita crear o modificar un rubro, dentro del *tab* de rubros del *wizard* para clientes se creó un link que lleva directamente al *wizard* que facilita dicha tarea.

Paginas para otras funcionalidades

Detalles del usuario seleccionado donde se puede observar los datos personales y los roles que tiene asociados, así como editar y asignar nuevos roles con el link de modificar. Se puede modificar todos los datos del usuario menos la contraseña (que es personal) y los datos de seguridad del mismo.

Fig. 107 Screenshots de páginas para otras funcionalidades 1

Parámetros del sistema donde se permitirá cambiar comportamientos del sistema y editar mensajes, descripciones y etiquetas correspondientes a cada empresa constructora que pida el producto. Estos parámetros se pueden agregar o modificar siendo la clave de cada entidad el nombre del parámetro, esto permite que si más adelante se desea modificar los comportamientos del sistema, se pueda realizar de manera muy sencilla agregando o modificando parámetros sin la necesidad de codificar requerimientos nuevos.

Edemas en la segunda imagen se puede observar como agregar, editar y eliminar roles del sistema.

Fig. 108 Screenshots de páginas para otras funcionalidades 2

En el siguiente par de imágenes se puede observar la sección de configuración de cuenta para un usuario *logueado* al sistema, en donde el mismo puede cambiar su contraseña, agregar un número telefónico que empieza con “+...”, activar la verificación en dos pasos y recordar el browser para evitar que la verificación en dos pasos salga cada vez que expira la sesión.

Fig. 109 Screenshots de páginas para otras funcionalidades 3

2.19. Aplicación móvil adicional

Como se había mencionado previamente en el anteproyecto uno de los motivos para crear el sitio de manera resposiva es la integración con phonegap, para generar una primera versión de la aplicación móvil.

Otra de las motivaciones a crear una app, es que se ha puesto muy de moda en los últimos años, sobre todo con el despegue de las posibilidades de los dispositivos móviles. Desde el nacimiento de la informática ha habido aplicaciones. Las apps sin embargo se refieren a un campo concreto, el de aplicaciones para móviles y dispositivos portátiles tipo *tablet*. Simplificando, las aplicaciones para ordenadores personales seguimos denominándolas por su nombre largo, mientras que usamos apps en su forma corta para referirnos a las pequeñas aplicaciones que se usan en dispositivos.

Para satisfacer las necesidades del cliente, se creó una app que complemente a la versión web y utilice la misma arquitectura web.

2.19.1. Desarrollo

Algunos de los aspectos más importantes a tener en cuenta sobre el desarrollo de la aplicación móvil de la empresa constructora son los siguientes.

Para desarrollar la app se usó:

- Cordova CLI
- Cordova android
- Sublime Text 3+
- Mac OSX 10
- Phonegap Build

1. Primero se generó una *keystore* nueva, abriendo un terminal y con el siguiente comando:
 - a. `keytool -genkey -v -keystore mikeystore.keystore -alias mi_alias -keyalg RSA -keysize 2048 -validity 10000S`
2. Se completaron los datos y pasos que son exigidos por el SDK de Google obteniendo como resultado dicho *keystore* que después se precisó para certificar la app en el store.
3. Después se abrió una cuenta de desarrollador en Google Play Store donde se tuvo que crear una nueva app, completando toda la información necesaria.
4. Creado el lugar en donde se subirá la aplicación y con el *keystore* ya generado, se procedió a crear una base que redireccione al sistema de la empresa constructora utilizando Sublime Text 3, compilando y probando mediante Phonegap Build.
5. Al final se quitó el certificado de *debug* de la .apk, se compiló utilizando el nuevo *keystore* y se subió la versión final al Play Store.

2.19.2. Sobre la herramienta

Phonegap Build es una solución que pensó el equipo de Nitobi (actual propietario junto con Adobe), bastante cómoda y útil para todo desarrollador móvil que quiera hacer uso de Phonegap, brindándole acceso a un compilador en la nube, donde se disponen una serie de herramientas que permiten el desarrollo de la aplicación móvil de forma sencilla y ágil, a través de métodos intuitivos que facilitan mucho el proceso de desarrollo.

Dando un poco de contexto sobre esta herramienta, la cual no desapareció cuando Nitobi fue adquirida por Adobe, sino todo lo contrario, se afianzó aún más, con mayor soporte, herramientas y creando el compilador en la nube que fue utilizado para desarrollar la app del proyecto.

Tal compilador en la nube recibe el nombre de PhoneGap Build, que, si bien este no hace parte del framework, es una herramienta útil para hacer uso de PhoneGap de una forma un poco más fácil sin la necesidad de instalar todo el SDK y entorno necesario para compilar una aplicación de Apache Cordova.

Es el complemento perfecto para todo lo que significa el *framework* también conocido anteriormente como Apache Cordova, el cual fue utilizado en el proyecto y gracias a esta poderosa herramienta resulta ser una tarea de baja complejidad realizar dicha app.

Esta herramienta es sin lugar a dudas un sistema muy bien construido, gracias al uso de la nube, donde se puede convertir las aplicaciones web en móviles, si bien el *framework* trabaja con siete plataformas, PhoneGap Build solo compila para seis: iOS, Android, BlackBerry OS, Symbian, Web OS y Windows Phone, integrado hace muy poco. Inicialmente en el proyecto solo se utilizó a modo de ejemplo el compilador de Android pero es totalmente posible estimando unos pocos días de desarrollo generar las aplicaciones de la empresa constructora para iOS y Windows Phone.

Cuando se registra en PhoneGap Build se puede acceder a todas las herramientas y todo el soporte necesario para compilar aplicaciones web en las plataformas móviles soportadas. Una vez dentro del sitio web, se puede crear las aplicaciones, donde se tiene la posibilidad de enviar el código a través de un repositorio git, o bien en un archivo .zip alojado localmente. Es importante destacar que este proyecto web debe tener un archivo index.html, pues este es el main o inicio de la aplicación, es posible darle el nombre e icono que va a usar la aplicación al ser instaladas en los diferentes sistemas operativos móviles.

Cuando se va a compilar la aplicación en el sitio web, se puede escoger la versión de PhoneGap, una vez terminado el proceso y subido el certificado para el respectivo App Store de la app, se da a “compilar” y en cuestión de minutos, PhoneGap Build avisa de posibles errores si los hay, de no ser así, se obtiene un paquete de instalación para cada uno de los sistemas operativos móviles que fueron mencionados algunas líneas más arriba.

2.19.3. Resultado obtenido

Gracias a las cualidades mencionadas del proyecto se pudo subir la app a Google Play en menos de dos días y brindarle al cliente un extra que facilite el acceso desde *smartphones*.



Fig. 110 Screenshot de app en Google Play



Fig. 111 Screenshot de la pantalla inicial

Pantalla inicial

Esta sección es la primera en ser visualizada por el cliente, en ella se muestra todos los rasgos principales de la constructora, sirve para mejorar el aspecto del front end y publicitar los rasgos mas sobresalientes de la constructora. Todos los atributos de esta sección son modificables por parametros desde el sistema de gestión integrada, dado que este producto se focalizo en ser multi empresa y generalizar lo mas posible los rasgos únicos que componen a cada constructora, gracias a esto se puede reutilizar el sistema con otras empresas que quieran un producto similar al ofrecido a ICAFAL. S.A.

Fue diseñada utilizando elementos HTML 5 del tipo sections, ya que estos ultimamente son muy utilizados para crear páginas responsivas y generar un efecto moderno y atractivo hacia el público en general.

Pantalla de login

Pantalla que permite el logeo de los usuarios al sistema, ya sean clientes, empleados o administradores a cada uno de los roles se los redireccionará a sus respectivas tareas principales, por ejemplo al cliente se lo redireccionará a la sección “perfil y tareas” mientras que al administrador se lo redireccionara a una pantalla que lo invitará a ingresar al modulo de sistema de gestión integrada.

Todas estas funcionalidades son igualmente accesibles mediante el servicio de login externo de Twitter, lo cual se requirirá que el cliente posea una cuenta de Twitter y que la vincule con una cuenta existente o que presione sobre el botón de login para registrarse con Twitter o si este ya había previamente vinculado que se loguee automáticamente con la cuenta asociada.

Fig. 112 Screenshot de la pantalla de login

Pantalla de perfil

Fig. 113 Screenshot de la pantalla de perfil

En esta pantalla se puede visualizar toda la información relativa al usuario *logueado*, las imágenes fueron programadas como links para más adelante utilizar un servicio *cloud* del tipo “bucket” que las provea y logre independizar al servidor de la carga de imágenes. Para configurar el perfil se agregó un link con una descripción donde se puede cambiar varios comportamientos del sistema para el usuario.

Además, se agregaron en 3 secciones diferentes los 3 tipos de tareas principales que el cliente puede realizar, “estimaciones”, “asesoramientos” y “construcciones”. Cada una encapsula y organiza las sub tareas que se pueden realizar en el sistema.

Cabe aclarar que los campos “EmailConfirmed” y “AccessFailedCount” son utilizados para mejorar la seguridad del sistema.

Sobre

Fig. 114 Screenshot de la pantalla de instrucciones

Pantalla de instrucciones

En este ejemplo de instrucciones para asesoramientos se puede observar una forma de motivación al cliente para utilizar el proceso de pedido de asesoramientos, donde se le explica de forma clara y sencilla los principales beneficios de realizar un pedido de asesoramiento mediante la forma web vs la forma tradicional.

Para ello se utilizó una pantalla previa a pedir un asesoramiento donde una serie de secciones remarcan las principales ventajas. Como el tiempo, el motivo y el resultado esperado.

Una de las cualidades del desarrollo de ésta sección fue que se pudo realizar en muy poco tiempo ya que se utiliza iconos de Font Awesome y código CSS provisto por las librerías de *front end* de Bootstrap.



Fig. 115 Screenshot de la pantalla de dashboard



Fig. 116 Screenshot de la pantalla de instrucciones para las estimaciones

Pantalla de dashboard

En esta pantalla se puede monitorear el estado general del sistema, se muestra graficas que vistas desde la versión móvil están ubicadas de manera vertical.

La primera muestra las cantidades de recursos, proyectos, obras y toda entidad del proyecto para brindar un pantallazo inicial del contenido del sistema.

La segunda es una gráfica de precios estimados en función del tiempo, esta brinda información sobre cómo fueron variando los costos generados por el sistema para todas las obras.

La tercera grafica muestra los metrajes realizados a los rubros, costos indirectos y subcontratos. Para visualizar fácilmente que es lo que más se utiliza en los proyectos de los clientes.

Pantalla de instrucciones para las estimaciones

El sistema de cálculo de costos y gestión de obras permite determinar el precio total detallado de una obra, el consumo de materiales y mano de obra por categorías. Para sintetizar este hecho, prácticamente se emplea un formato denominado estimación de obra, el cual es el documento que nos permite llevar el control financiero de una obra y en el cual se resumen los conceptos, cantidades de obra ejecutada y sus importes para un periodo de tiempo determinado, así mismo, es el documento que permite al contratista presentar su requerimiento de cobro al Contratante con objeto de contar con la solvencia necesaria y no afectar el programa general de ejecución de la obra. Para ello se partirá de la siguiente información.

Para ello se partirá de los pasos provistos en ésta pantalla, los cuales resumen a grandes rasgos su función en el proceso de estimación de costos.

The screenshot shows a mobile application interface for selecting clients. At the top, there's a header with the company name 'IGAFAL S.A.' and some navigation icons. Below the header, a title bar says 'Cliente - Seleccionar'. A note below the title explains the purpose: 'Listado de todos los usuarios del sistema a los cuales se les puede inicializar un proceso de estimación. Presiona en continuar sobre alguno de los usuarios para trabajar con los proyectos y asesoramientos de dicho usuario que actuará como cliente. Recuerda que puedes filtrar y ordenar por cualquier columna para facilitar la tarea.' There's a search bar labeled 'Buscar'. Below the search bar is a table with the following columns: Nombre, Apellidos, PhoneNumber, and País. The table contains ten rows of data. Each row has a 'Continuar' button at the end. At the bottom of the table, it says 'Mostrando 1 a 10 de 10 filas' and has navigation buttons for 'Previous', '1', and 'Next'.

Fig. 117 Screenshot de la pantalla inicial del wizard clientes

Pantalla inicial del wizard clientes

En esta pantalla se puede visualizar un listado de todos los usuarios del sistema a los cuales se les puede inicializar un proceso de estimación. Presionando sobre continuar sobre alguno de los usuarios, se podrá trabajar con los proyectos y asesoramientos de dicho usuario que actuará como el rol de cliente.

Esta pantalla al igual que todos los *grids* del sistema se pueden filtrar y ordenar por cualquier columna para facilitar la tarea del usuario.

Se intentó agregar lo mínimo e indispensable para identificar a un usuario en el *grid* y agregar la menor cantidad de columnas posibles para mejorar la visualización móvil del sitio.

The screenshot shows a mobile application interface for customizing a project. At the top, there's a header with the company name 'IGAFAL S.A.' and some navigation icons. Below the header, a title bar says 'Personalizar obra'. A note below the title explains the purpose: 'Sección autogenerada a partir de los procedimientos constructivos de los rubros definidos para ésta obra. Puedes clickear directamente sobre el nombre del rubro para ver y modificar su contenido.' There's a 'Modificar rubros +' button. Below the note, there are three panels representing different materials: 'Cocina marmol', 'Ventanas aluminio', and 'Cocina madera'. Each panel shows a timeline with steps: 'Paso: 1', 'Paso 2', and 'Paso 3'. Each panel also includes a note about dates and quantities.

Fig. 118 Screenshot de la pantalla de personalizar obra del wizard clientes

Pantalla de personalizar obra del wizard clientes

Esta sección es autogenerada a partir de los procedimientos constructivos de los rubros definidos para ésta obra. Se puede *clickear* directamente sobre el nombre del rubro para ver y modificar su contenido.

Se listan ordenados por la fecha que se asignó al realizar el metraje del rubro, donde para cada rubro se inicializó una serie de pasos o procedimientos constructivos para realizar su construcción.

Gracias a ésta estructura se puede generar el procedimiento constructivo para una obra entera a partir de los "sub" procedimientos de cada rubro. En cada panel dentro del *timeline* se puede observar el nombre del rubro, las fechas, el metraje, la unidad en la que se expresa el rubro y los pasos a seguir.

Pantalla de pedido de asesoramientos

En esta pantalla es donde se pide un asesoramiento desde la vista del cliente, éste deberá seguir una serie de pasos donde el primero será describir el proyecto con, nombre, país y zona para ayudar a identificar las obras que se asignarán al proyecto.

Después se completarán las cantidades y metros cuadrados que se deseen para el proyecto.

En el siguiente paso se seleccionará los haberes que se deseen para el proyecto y en el último paso se deberá describir la obra brindando todos los detalles posibles para mejorar la estimación final que se le enviará al cliente.

Al culminar el proceso se le enviará una notificación al administrador y éste creará la obra con los detalles provistos anteriormente y enviará una notificación al cliente.

Fig. 119 Screenshot de la pantalla de pedido de asesoramientos

Pantalla de noticias

Las noticias son una buena sección para mostrar las novedades de la construcción o sobre lo que el cliente prefiera informar a los usuarios que visiten el sitio de la constructora.

La misma se controla desde el sistema de gestión integrada y se visualiza para cualquier usuario del sistema (autenticado o no autenticado) en una serie de tarjetas horizontales, que se acomodan dinámicamente al ancho de la página y al tamaño de las otras tarjetas a su alrededor.

Dicho lo anterior esto, la idea de presentar las noticias vistas como tarjetas dinámicas mejoró a la integración con la versión móvil del sitio.

Fig. 120 Screenshot de la pantalla de noticias

2.20. Performance y usabilidad

Se realizó un estudio de algunas recomendaciones de expertos en el uso de las normas ISO 9241 relacionada a usabilidad de los sistemas, y recomendaciones de la W3C, el cual es un organismo internacional de normalización del desarrollo web.

Solo se tomaron en cuenta los aspectos más relevantes con los que debe cumplir un sitio web de una empresa u organización, de forma tal que se logre llegar al usuario y se logre brindar la comodidad esperada en la navegación del sitio.

Existen en Internet muchos servicios para consultar la optimización de un sitio, por varios factores que afectan a su velocidad de carga. Son servicios que nos sirven para saber si una web está o no bien hecha en cuanto a optimización de su velocidad, usando varios factores generales que afectan a la rapidez de carga. Pero no solo nos permite saber qué se puede mejorar, sino que además nos ofrecen referencias excelentes para saber qué es lo que deberíamos hacer para mejorar allí donde sea posible.

Muchos de esos factores dependen de la configuración del servidor donde el sitio está alojado, pero muchos otros dependen también de la optimización del código de la web, atendiendo no solo al HTML, sino también al CSS, JavaScript y las imágenes u otros recursos externos. Por lo tanto, seguro que por un lado o por otro tendrás tareas y material para poder trabajar sobre el sitio, independientemente de si controlamos o no el servidor donde está alojada.

Hoy la velocidad de carga es un factor fundamental, no solo para mejorar la experiencia de usuario, sino también el posicionamiento en buscadores. Sabemos que si un usuario espera demasiado para consultar tu sitio acabará buscando otras alternativas y consultando los recursos de tu competencia. En este sentido pocos segundos son vitales para el éxito de un negocio en Internet. También se sabe que Google usa la velocidad de carga de los sitios como un factor para definir su calidad y por tanto también su posición entre los resultados de las búsquedas en su buscador.

"Si Amazon tardase 0.1 segundos (la décima parte de un segundo) más en cargar, perdería el 1% de sus ventas. Esto nos da una idea de hasta qué punto una ridícula mejora en la carga del sitio puede ser absolutamente representativa en cuanto a ventas u obtención de ingresos.

Para analizar el proyecto en la nube de Azure se utilizó pingdom.com, Pingdom Tools es una web gratuita que analiza la velocidad de carga de tu sitio web y te propone recomendaciones para hacerla más rápida. Analiza que elementos son rápidos, lentos, que buenas prácticas no se están siguiendo y muestra los resultados de manera que pueden ser útiles tanto para *webmaster* expertos como novatos.

Las principales características son:

- Examina todas las partes de la web: Tamaños de archivos, tiempos de carga, y otros detalles de cada uno de los elementos que forman la web (HTML, JavaScript, css, imágenes, etc.). Una vez analizados muestra un listado que se puede ordenar por tiempos para identificar los cuellos de botella.
- Descripción general del rendimiento: Automáticamente analiza una gran cantidad de estadísticas para obtener el resultado de la prueba.
- Grado de rendimiento y consejos: Muestra cómo se adapta nuestra web a las buenas prácticas marcadas por Google Page Speed o Yahoo YSlow, además muestra algunos consejos sobre cómo mejorar la velocidad de carga.
- Histórico de pruebas: Almacenan cada una de las pruebas que se realicen y muestras las diferencias entre cada una de ellas. Muy útil para analizar si los cambios están surtiendo el efecto deseado.
- Prueba en varias ubicaciones: Muestra las diferencias de cada de la web en Europa, Estados Unidos.
- Compartir los resultados: Permite descargarse los resultados para compartirlos o analizarlos con detenimiento.

El resultado fue el siguiente:

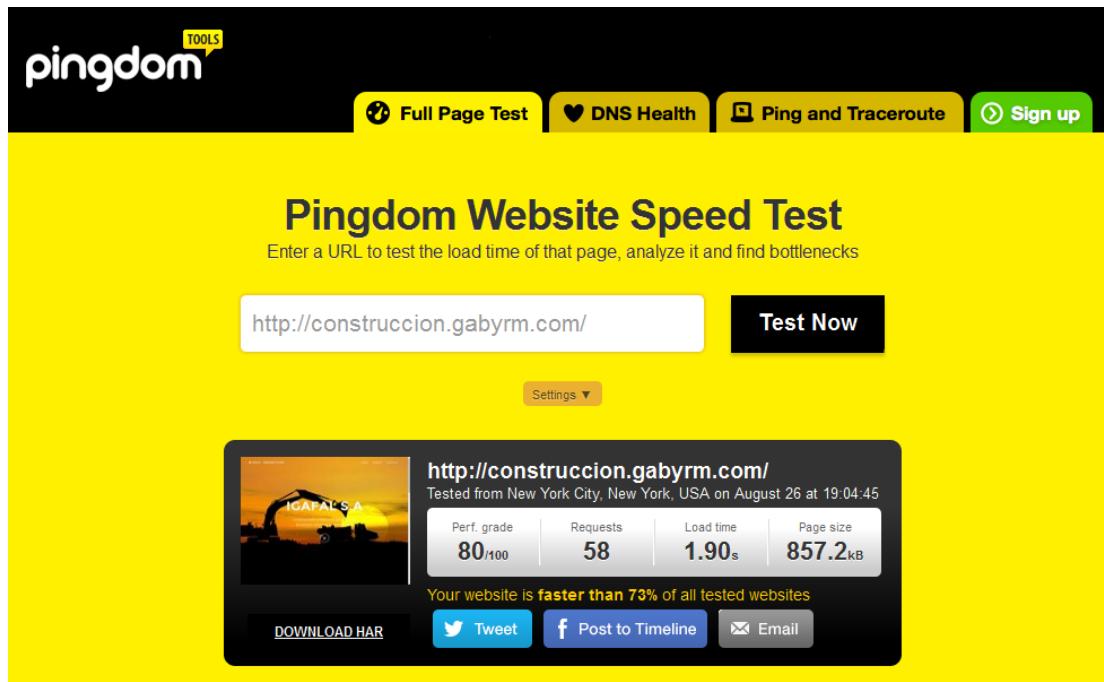


Fig. 121 Screenshot de PingDom website speed test

Lo primero que muestra es un resultado global en el que dice que nota sobre 100 tiene tu web, en mi caso casi notable 80/100, además de las *request* realizadas (58), el tiempo de carga (1,90sg) y el tamaño (857,2 kb). Además, muestra como es de rápida tu web respecto al resto de sitios analizados, en mi caso es más rápida que el 73% de web analizadas.

Un poco más abajo, muestra un listado que se puede modificar porque tiene varias pestañas. En la pestaña *waterfall* muestra que elementos ha cargado, su tamaño y tiempo de carga.

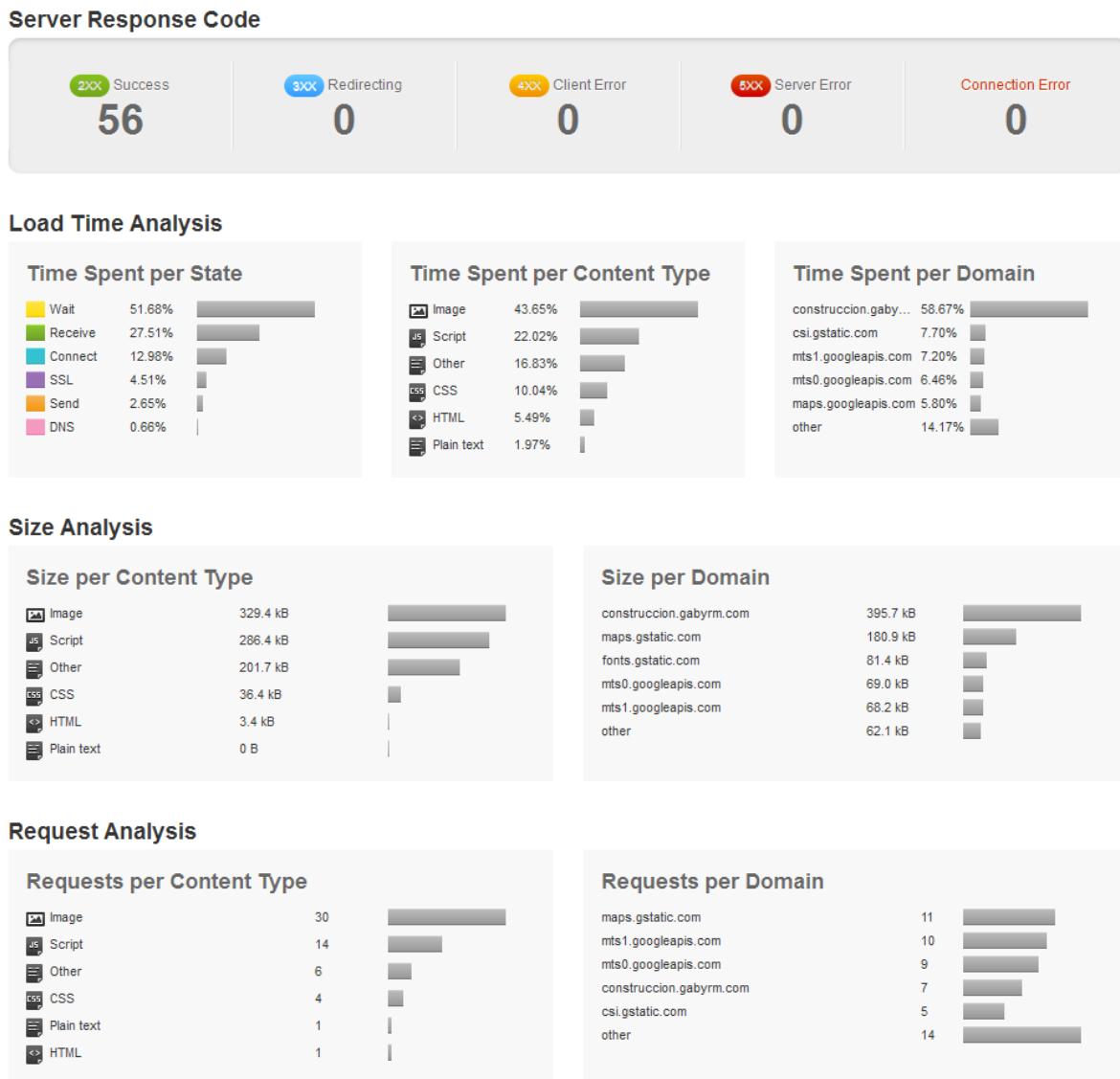


Fig. 122 Screenshot de PingDom load time analysis

Para cumplir con el requerimiento no funcional cuyo identificador es “RNF5” que exige al sistema ser estable en todo momento.

Se utilizó una herramienta de PingDom que ayuda a tomar estadísticas para comprobar que el sistema llega a un mínimo del 95% del tiempo de ejecución sin caídas por cada mes utilizando.

Esta herramienta realiza un ping cada 2 minutos, el cual se promedian los paquetes que llegan al servidor exitosamente, cuáles no y en qué tiempo lo hacen para después con la información recolectada, generar un porcentaje de estabilidad y tiempo de respuesta general del sistema.

<u>October 2015</u>	100%	341 ms
<u>September 2015</u>	100%	331 ms
<u>August 2015</u>	99.90%	382 ms
<u>July 2015</u>	100%	378 ms
<u>June 2015</u>	96.33%	391 ms
<u>May 2015</u>	100%	334 ms

Fig. 123 Screenshot de PingDom website history

2.21. Conclusiones finales

2.21.1. Introducción

Después de concluir todos los *sprints*, cumpliendo con los plazos y productos entregables definidos surge la necesidad de evaluar y analizar de forma retrospectiva lo ocurrido durante el transcurso del proyecto.

Esto se pudo lograr gracias a una constante interacción con el cliente y a su vez con los tutores del proyecto, pudiendo establecer un ritmo de trabajo enmarcado en tareas y documentando simultáneamente en cada sprint lo ocurrido. A continuación, se analizará utilizando lo documentado, se describirán los puntos más importantes aprendidos y se plantearán posibles las posibles mejoras y conclusiones.

2.21.2. Lecciones aprendidas

Las principales lecciones que dejó el proyecto, fueron las siguientes.

Del proyecto a la solución real: se identificaron problemas “comunes” que por temas académicos y de tiempo no se suelen dar durante el transcurso de la carrera. Estos tienen un papel importante que le genera un valor agregado al proyecto que se le brindará al cliente. A pesar de que no fueron vistos durante la carrera se posee con los conocimientos y herramientas necesarias como para aprender, incorporarlos y alcanzar las expectativas tanto del cliente como las académicas.

Uno de ellos fue la constante reinención y utilización de lo último que brinde la tecnología dado que no existen limitantes en ese aspecto para este proyecto en particular, en éste caso el *framework* .NET donde utilizarlo de dicha manera implicó un constante aprendizaje e investigación ya sea desde conceptos básicos de implementación hasta mejores formas de abarcar soluciones a problemas más complejos.

La búsqueda de una solución que brinde la mayor seguridad posible en el menor tiempo de implementación. Intentando lograr un sistema que sea funcional pero que a su vez también tenga un mínimo de seguridad que prevea contra los ataques y problemas más comunes que suelen suceder en proyectos ya sea a corto o media plazo. Para esto fue fundamental analizar antes de programar, investigando todas las alternativas posibles y escoger la que mejor se adapte a las necesidades del cliente. En este caso se optó por utilizar una combinación que proveía el patrón de MVC de .NET, *tags* de seguridad, controladores y otras recomendaciones que se fueron aprendiendo e implementando durante el desarrollo del proyecto.

Por otro lado, la constante atención a temas que son puramente de diseño y la búsqueda de cómo adaptar la tecnología automatizando todo aquello que sea rutinario, reutilizando y adaptando según se necesite de manera que se cree un “estándar” o marco de trabajo que facilite y ayude progresivamente a realizar una tarea que lleva un tiempo de trabajo para la cual no se poseía el tiempo necesario. Para esto se utilizaron varias librerías JavaScript que dada la naturaleza del patrón arquitectónico escogido no se tuvieron grandes inconvenientes en su integración y en conjunto con lenguajes de sintaxis como Razor C# se logró trabajar con naturalidad y mantener una delegación de responsabilidades que faciliten la tarea de programación.

Dicho esto, a pesar de tener una curva de aprendizaje un poco más lenta que si se hubiese optado por otras soluciones, se obtuvo el beneficio de aprender, incorporar nuevos conocimientos y utilizar marcos de trabajo que son comprobados y utilizados a gran escala globalmente.

Las problemáticas mencionadas anteriormente tienen un papel académico importante y son esenciales para la realización profesional, fue necesario implementar dichos puntos multiplicando las tareas a realizar. Esto consumió tiempo y esfuerzo que no fue previsto, pero que si se tuviese que realizar una estimación en base a la experiencia obtenida se llegaría a una estimación más acertada y acorde a las necesidades del cliente.

De la necesidad del cliente a la innovación: Esto fue un punto muy importante que surgió del relevamiento de las necesidades principales del cliente, que implicó manejar los requerimientos apuntando a la innovación y los atributos que actúen como “diferenciales” del proyecto cumpliendo con todas las obligaciones al mismo tiempo.

Por ejemplo una de ellas fue partiendo de la necesidad principal de estimar obras dado el cumplimiento (con la mayor prioridad posible) se buscó desarrollar un entorno que aproveche lo desarrollado y exponga a los futuros usuarios todas las ventajas que brinda la empresa constructora al estimar obras. Teniendo el consentimiento del cliente y la aprobación de lo desarrollado se relevaron nuevos requerimientos que permitieron desarrollar un sistema que cumpla con lo propuesto y que a su vez cumpla con las expectativas académicas y personales.

De la adversidad a la ingeniería de software: Durante el transcurso del proyecto se comprobó la importancia que tiene aplicar los conocimientos aprendidos en la carrera y como impactan en el desarrollo. Cada problema que se encontraba tenía una razón que junto a buenas prácticas y técnicas vistas en la ingeniería de software se pudieron resolver con efectividad y prevenir a futuro.

La planificación del Proyecto tuvo gran importancia y estas buenas prácticas aportaron al cumplimiento de los objetivos. Si bien algunas tareas planificadas en los *sprints* durante el transcurso del anteproyecto tuvieron algunas diferencias en lo que respecta al esfuerzo necesario para implementar los requerimientos se pudo priorizar aquellas que le eran de más importancia para el cliente y gestionar aquellas con menor prioridad, documentando en cada caso los desvíos necesarios.

De la metodología de desarrollo a la realidad: Aplicar la metodología de desarrollo “scrum” permitió que el proyecto avanzara acorde a lo que el cliente esperaba, logrando obtener resultados cada vez mejores utilizando tecnologías que en gran parte eran desconocidas. Esto aporto en disminuir el tiempo en cada avance de sprint, dar un marco de trabajo que divida tareas y permita la revisión y defina reuniones de retrospectiva para evaluar resultados, manteniendo un control continuación de la evolución del proyecto conforme pasaba el tiempo.

2.21.3. Posibles mejoras

Debido a que el proyecto tiene un tiempo acotado y algunas de los requerimientos de baja prioridad no pudieron ser implementados en su totalidad, se puede definir una lista de mejoras y de requerimientos a mejorar.

Se ha establecido durante las diferentes charlas con el cliente sobre lo que debería o no de tener el sistema. Y lo cierto es que establecer un plan de mejoras de sistemas es clave para crecer en la empresa, de manera que los encargados del soporte no sólo deban ocuparse del mantenimiento sino de aportar a la empresa. Para ello una de las acciones que se podrían tomar es integrar un *desk* de ayuda al usuario, dónde se pueda monitorear, administrar tickets y registrar distintas ocurrencias del sistema. Lo cual brindará al usuario una forma para buscar información y soluciones a los problemas más comunes y como solucionarlos reutilizando otros tickets reportados.

Además, se quiere que no sólo el sistema funcione, sino que lo haga bien, y esta es una de las principales prioridades junto con la automatización de procesos, intentar que la intervención de los usuarios en los mantenimientos sea la menor posible, con esto se quiere llegar antes al problema que el usuario, para ello se creo mediante una “lluvia de ideas”, ideas para posibles mejoras. Algunas de ellas son las siguientes.

- Mejora de interfaces en base a opiniones de usuarios del sistema.
- Mejora de procesos, tomando más variables a tener en cuenta en la estimación.
- Exponer estimaciones como *webservices* para su posible reutilización en otros sistemas.
- Integrar la librería de JQuery para el *tooltip* de calendario en la interfaz gráfica.
- Integrar Ajax en el body para evitar el *refresh* de todo el *layout*.
- Integrar un desk de ayuda al usuario y apertura de tickets, posiblemente ZenDesk.
- Integrar *push notifications* para dispositivos móviles utilizando Cordova o Parse.
- Integrar un Global.resx para el soporte multilenguaje del sistema (inicialmente inglés).
- Integrar a los empleados como parte del cálculo de la estimación.
- Integrar un *bucket* u otro *cloud service* para el servido de imágenes.
- Integrar un *webservice* que actualice automáticamente el precio de los recursos.
- Generar conversión de estimación en diferentes monedas.
- Generar un .pdf para el usuario con la estimación reutilizando el HTML de la estimación.
- Corregir y probar en diferentes resoluciones móviles aspectos de diseño.
- Corregir textos y mensajes en general que sean poco explicativos.

2.21.4. Desarrollos a futuro

Empezando por la tabla de cumplimiento de requerimientos definido en el punto 2.10 de la documentación, para aquellas funcionalidades que quedaron por implementar se podría volver a estimar y asignar una prioridad e implementarlas en un sprint previo al sprint en el cual se realizarían las mejoras, de forma tal que antes de dedicarse a mejorar el sistema se terminen aquellas funcionalidades que quedaron pendientes y una vez se termine, comenzar con *sprints* que estén dedicados exclusivamente a mejorar el comportamiento del sistema.

Algunos de los desarrollos a futuro que quedaron pendientes son los siguientes.

Exponer cargos disponibles en la empresa: Donde se mostraría cuando existan cargos disponibles en la empresa, estos se podrán exponer en la fachada publicitaria, permitiendo que los interesados envíen sus formularios y posteriormente si cumplen con los requerimientos agendar una entrevista.

Calcular la paramétrica para una obra que no sea de este año: En éste requerimiento que quedó pendiente se debería permitir que para una obra que no sea de este año se pueda actualizar su precio mediante las paramétricas del ministerio de transporte y obras públicas. Esto ayudaría a que las obras del sistema sean reutilizables.

Cronograma físico: Registrar cuando se deberían completar las construcciones de los rubros ayudaría a mantener un mayor control sobre el avance de la obra y se podría integrar perfectamente con la sección del *wizard*, agregando un valor extra al proyecto con poco esfuerzo y tiempo de implementación.

2.21.5. Conclusiones

Al finalizar el proyecto se llegó a la conclusión que se cumplieron con los objetivos generales del trabajo, tanto desarrollando un producto real como en el aprendizaje académico.

La metodología de trabajo aplicada resultó ser una alternativa totalmente viable para este tipo de proyectos y ayudo paulatinamente a que el proyecto siguiera un ritmo de trabajo constante en manteniendo un nivel entre el desarrollo y la planificación en comparación con metodologías más tradicionales. La cual si se sigue de manera correcta y se respeta los plazos y tiempos de trabajo estipulados se puede cumplir con las entregas de cada sprint. Cumplir con los lineamientos fue de vital importancia ya que se trabaja 8 horas y mantener un orden y un seguimiento diario al proyecto permitió manejar las dos responsabilidades equitativamente.

Por último, una conclusión gratificante es saber que el producto entregado permitió que el cliente esté satisfecho y que finalmente le aportará en el desarrollo de su empresa.

Glosario

- **Phonegap** - es un *framework* para el desarrollo de aplicaciones móviles producido por Nitobi, y comprado posteriormente por Adobe Systems.^{3 4} Principalmente, PhoneGap permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3.
- **ASP.NET** - es un *framework* para aplicaciones web desarrollado y comercializado por Microsoft.
- **Framework** - En el desarrollo de software, un *framework* o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.
- **Licitación** - es el procedimiento administrativo para la adquisición de suministros, realización de servicios o ejecución de obras que celebren los entes, organismos y entidades que forman parte del Sector Público.
- **Azure** - es una plataforma general que tiene diferentes servicios para aplicaciones, desde servicios que alojan aplicaciones en alguno de los centros de procesamiento de datos de Microsoft para que se ejecute sobre su infraestructura (Cloud Computing) hasta servicios de comunicación segura y federación entre aplicaciones.
- **Análisis FODA:** metodología de estudio de la situación de una organización o un proyecto, analizando sus características internas (Debilidades y Fortalezas) y su situación externa (Amenazas y Oportunidades).
- **IDE:** Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de *Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios de ellos.
- **Hosting:** (alojamiento o también conocido como hospedaje web, alojamiento web, *web site hosting*, *web hosting* o *webhosting*) es un negocio que consiste en alojar, servir, y mantener archivos para uno o más sitios web.
- **Deployment:** procedimiento necesario para la instalación y puesta en funcionamiento de una aplicación (incluyendo la base de datos, *front end*, servicios o configuración de servidores).
- **Open Source** (Código abierto): es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.
- **Plan de SCM (Software Configuration Management Plan o Gestión de Configuración de Software):** Plan de gestión que contempla las políticas de configuración del software. Establece cómo se coordinarán las actividades como la elaboración de código fuente por varios desarrolladores simultáneamente, el seguimiento del estado de las versiones y sus cambios y la conducción de la integración de las partes del software en un solo producto de software.

- **Plan de SQA** (*Software Quality Assurance Plan* o Plan de garantía de calidad del Software): plan que organiza el desarrollo del software con el fin de que el proceso de creación de este siga determinados estándares que aseguren la calidad del resultado.
- **UML** (*Unified Modelling Language* o Lenguaje Unificado de Modelado): lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.
- **Scrum**: es un marco o metodología de trabajo para la gestión y desarrollo de software que se basa en un proceso incremental e iterativo, utilizado comúnmente en entornos basados en el desarrollo ágil de software.
- **Sprint**: en la metodología scrum, es el término que denomina a una iteración que está acotada en el tiempo, usualmente entre 2 y 4 semanas, durante la cual el Equipo trabaja para convertir los requerimientos del usuario en un incremento del producto potencialmente productivo.
- **Droplet** - DigitalOcean llama a sus servidores virtuales, gotitas (Droplets); cada gota que se crea es un nuevo servidor virtual para su uso personal.

Lista de referencias bibliográficas

- [1] I. F. Sommerville, “Administración de un proyecto ágil” en *Ingeniería de Software*, 9na.ed., Fife, Escocia: University of St. Andrews, 2011, pp. 72-74.
- [2] R. S. Pressman, “Identificación de riesgos” en *Ingeniería de Software: Un enfoque práctico*, 7ma.ed., Nueva York, Estados Unidos: University of Connecticut, 2010, pp. 642-645.
- [3] R. S. Pressman, “El aspecto administrativo” en *Ingeniería de Software un enfoque práctico*, 7ma.ed., Nueva York, Estados Unidos: University of Connecticut, 2010, pp. 555-562.
- [4] Inteco, (2008, Dec.). Guía Avanzada de Gestión de Requisitos [Online], pp. 18-63. Disponible: <https://www.incibe.es/file/TnOlvX7kM5rlBliD4wnTxQ>

Bibliografía

- FOWLER, Martin, et. al. 1999. *UML Gota a gota*. México: Prentice Hall.
- PALACIO, Juan. 2008. *Flexibilidad con Scrum. Adaptando los Procesos a la Empresa. Principios de diseño e implementación de campos de Scrum*.
- PRESSMAN, Roger, 2010, *Ingeniería del software: Un enfoque práctico* (7ma.ed), University of Connecticut.
- SOMMERVILLE, Ian, 2010, *Ingeniería de software* (9na.ed), University of St. Andrews.