

Sistema de Administración de Consorcios (SAC), caso de estudio Laravel

1- Candidatos

- Arizaga Alejandro
- Fernandez Gastón

2- Introducción

SAC (Sistema de Administración de Consorcios) se centrará en la liquidación de expensas y feedback Administrador-consorcistas, Este va a ser web. Desarrollado en Laravel con Base de datos en MariaDB.

Como no poseemos conocimientos sobre el framework Laravel 5.3 decidimos centrarnos en el estudio de dicho framework y a su vez el mercado necesita un software para la administración de propiedades horizontales, que brinde las herramientas necesarias para dicha función, facilite la comunicación y control del copropietario.

Decidimos utilizar Laravel porque:

- Está diseñado para desarrollar bajo el patrón MVC (modelo - vista - controlador), centrándose en la correcta separación y modularización del código.
- Integra un sistema ORM de mapeado de datos relacional llamado Eloquent, aunque también, permite la construcción de consultas directas a base de datos mediante su Query Builder.

- Utiliza un sistema de plantillas para las vistas llamado Blade, el cual hace uso de la cache para darle mayor velocidad. Blade facilita la creación de vistas mediante el uso de layouts, herencia y secciones.

Se elige MariaDB como motor de base de datos debido a que es un motor gratuito y relacional en lenguaje MySQL

Tabla 1: Análisis FODA

Fortalezas	Oportunidades	Debilidades	Amenazas
El framework es MVC	Hay poca competencia	Framework desconocido	La mayoría de los administradores se resisten al cambio
El framework se programa en PHP que es basado en C	La mayoría de los administradores tercerizan servicios, lo que podrían tomar como ahorro de dinero al centralizar servicios	Falta de experiencia en uso de frameworks	Cambios en la legislación vigente
Usa Base de datos relacionales en SQL	Cambio inminente en los datos contenidos en la liquidación de expensas, por medio de la futura reglamentación de ley provincial 14701	Desconocimiento de herramientas Git y uso de repositorios	Puede que algún usuario no tenga acceso a internet
	Información centralizada de datos de proveedores y administradores		
	Abolición de ley 13512: Régimen legal de la propiedad horizontal		

Según lo plasmado en el análisis FODA del proyecto consideramos que el mismo es viable.

Nos basaremos en el “Código civil y comercial capítulo V”, la futura promulgación de la ley provincial 14701, la documentación oficial de Laravel, la documentación oficial de MariaDB, Documentación oficial de PHP, utilizando los

libros “UML 2 Iniciación, ejemplos y ejercicios corregidos [3ª edición]” para la elaboración de la documentación a presentar, el “Manual práctico de propiedad horizontal” nos dará principios básicos para la administración de propiedades horizontales, el libro *Ingeniería de Software* de Sommerville Ian no servirá como guía para la realización de todo el proyecto completo.

3- Planeamiento del método

Utilizaremos una metodología Scrum porque es una metodología en la que cada miembro del equipo trabaja en forma individual y porque una vez que el sprint esta funcionando y aprobado no se vuelven a tocar en ningún momento. Dando al equipo, de esta forma, la posibilidad de trabajar cuando los tiempos laborales y personales lo permitan y la precisión de saber que lo que está terminado ya queda

Presentaremos los siguientes diagramas:

- Diagrama de casos de uso
- Diagrama Entidad Relación
- Diagrama de Clases
- Diagrama de componentes de software y sus interrelaciones.
- Product backlog
- Sprint backlog
- Historias de usuario

Utilizaremos para la realización del software a entregar Git y planificador de tareas Trello

Utilizaremos un esquema de versionados con tres ramas principales:

- Desarrollo

- Testing
- Producción: Esta rama será para lo que esté listo para pasar a producción

4- Composición general tentativa

Introducción

Capítulo 1 Planeamiento del problema

- 1.1 Objetivos
- 1.2 Justificación
- 1.3 Estudio de factibilidad
- 1.4 Alcances y limitaciones
 - 1.4.1 Alcances
 - 1.4.2 Limitaciones

Capítulo 2 Marco teórico

- 2.1 Bases teóricas
 - 2.1.1 Arquitectura cliente servidor
 - 2.1.1 Protocolo HTTP
 - 2.2.2 Arquitectura MVC
 - 2.2.3 Lenguaje PHP
 - 2.2.4 Framework Laravel
- 2.2 Bases legales
 - 2.2.1 Código civil- artículos 2037 al 2072
 - 2.2.2 Ley provincial 14701
- 2.3 Definición de términos básicos

Capítulo 3 Diseño tecnológico

- 3.1 Síntesis descriptiva del proyecto
- 3.2 Tipo de Proyecto
- 3.3 Técnicas e instrumentos de recolección de datos
- 3.4 Cronograma de ejecución de actividades
- 3.5 Análisis de los requerimientos
- 3.6 Estructura de la base de datos
- 3.7 Diagrama de clases
- 3.8 Diagrama de componentes
- 3.9 Modelo entidad relación
- 3.10 Product backlog
- 3.11 sprint backlog
- 3.12 Historias de usuario

Conclusiones

Bibliografía

Anexos

5- Referencias preliminares

Laravel documentación oficial: <https://laravel.com/docs/5.3/releases>

MariaDB documentación oficial: <https://mariadb.com>

Ley provincial 14701: <http://www.gob.gba.gov.ar/legislacion/legislacion/l-14701.html>

Infojus. (2014). *Código civil y comercial. Libro Cuarto. Derechos Reales*. Argentina:

Autor

Carlos Diego Calvo. (1999). *Manual práctico de propiedad horizontal*. Buenos Aires:

Editorial Universidad

Laurent Debrauwer y Fien van der Heyde (2005). *UML 2 Iniciación, ejemplos y ejercicios corregidos [3ª edición]*. España: Editorial ENI

Sommerville Ian. (2011). *Ingeniería de Software*. México: Pearson

6- Cronograma

Inicio	Hito	Duración	Fin	Responsable
01/07/2016	Adquisición de requerimientos	8	31/07/2016	Fernandez
01/08/2016	Especificación de requerimientos	5	31/08/2016	
01/09/2016	Confección tentativa del DER	2	15/09/2016	
15/09/2016	Confección de Casos de Uso	2	30/09/2016	Fernandez y Arizaga
01/12/2016	Aprendizaje inicial	26	31/12/2016	
02/01/2017	Instalación del ambiente de desarrollo	1	02/01/2017	
03/01/2017	CU 20	2	04/01/2017	Fernandez
05/01/2017	Documentación CU 20	2	06/01/2017	
07/01/2017	Testing CU 20	4	11/01/2017	
12/01/2017	CU 17	1	12/01/2017	Arizaga
13/01/2017	Documentación CU 17	1	13/01/2017	
14/01/2017	Testing CU 17	1	14/01/2017	
16/01/2017	CU 18	1	16/01/2017	Fernandez
17/01/2017	Documentación CU 18	1	17/01/2017	
18/01/2017	Testing CU 18	1	18/01/2017	
19/01/2017	CU 3	4	23/01/2017	Arizaga
24/01/2017	Documentación CU 3	2	25/01/2017	
26/01/2017	Testing CU 3	6	01/02/2017	
02/02/2017	Testing Integral Sprint 1	2	03/02/2017	Fernandez
04/02/2017	CU 2	4	08/02/2017	Arizaga
09/02/2017	Documentación CU 2	2	10/02/2017	
11/02/2017	Testing CU 2	2	13/02/2017	
14/02/2017	CU 5	4	17/02/2017	
18/02/2017	Documentación CU 5	2	20/02/2017	
21/02/2017	Testing CU 5	6	27/02/2017	
28/02/2017	Testing Integral Sprint 2	2	01/03/2017	
02/03/2017	CU 1	4	06/03/2017	
07/03/2017	Documentación CU 1	2	08/03/2017	
09/03/2017	Testing CU 1	6	15/03/2017	Fernandez
16/03/2017	CU 4	1	16/03/2017	
17/03/2017	Documentación CU 4	1	17/03/2017	
18/03/2017	Testing CU 4	2	20/03/2017	

21/03/2017	CU 12	1	21/03/2017	
22/03/2017	Documentación CU 12	1	22/03/2017	
23/03/2017	Testing CU 12	2	24/03/2017	
25/03/2017	CU 13	1	25/03/2017	
27/03/2017	Documentación CU 13	1	27/03/2017	
28/03/2017	Testing CU 13	2	29/03/2017	
30/03/2017	Testing Integral sprint 3	2	31/03/2017	
01/04/2017	CU 9	4	06/04/2017	Arizaga
03/04/2017	1° reunión presentación de avances	1	04/04/2017	
07/04/2017	Documentación CU 9	2	08/04/2017	Arizaga
10/04/2017	Testing CU 9	6	15/04/2017	
17/04/2017	Testing Integral sprint 4	2	18/04/2017	
19/04/2017	CU 6	4	22/04/2017	
24/04/2017	Documentación CU 6	2	25/04/2017	
26/04/2017	Testing CU 6	6	02/05/2017	
03/05/2017	CU 15	1	03/05/2017	Fernandez
04/05/2017	Documentación CU 15	1	04/05/2017	
05/05/2017	Testing CU 15	2	06/05/2017	
08/05/2017	CU 7	1	08/05/2017	
09/05/2017	Documentación CU 7	1	09/05/2017	
10/05/2017	Testing CU 7	2	11/05/2017	
12/05/2017	Testing Integral sprint 5	2	13/05/2017	
15/05/2017	CU 14	1	15/05/2017	
16/05/2017	Documentación CU 14	1	16/05/2017	
17/05/2017	Testing CU 14	2	18/05/2017	Arizaga
19/05/2017	CU 19	1	19/05/2017	
20/05/2017	Documentación CU 19	1	20/05/2017	
22/05/2017	Testing CU 19	1	22/05/2017	Fernandez
23/05/2017	CU 10	6	29/05/2017	
30/05/2017	Documentación CU 10	2	31/05/2017	
01/06/2017	Testing CU 10	9	10/06/2017	
12/06/2017	CU 8	2	13/06/2017	
14/06/2017	Documentación CU 8	2	15/06/2017	
16/06/2017	Testing CU 8	4	20/06/2017	
21/06/2017	Testing Integral sprint 6	2	22/06/2017	Arizaga
23/06/2017	CU 11	2	24/06/2017	Fernandez
26/06/2017	Documentación CU 11	2	27/06/2017	
28/06/2017	Testing CU 11	4	01/07/2017	
03/07/2017	2° reunión Presentación de avances	1	03/07/2017	
04/07/2017	CU 16	2	06/07/2017	Arizaga
07/07/2017	Documentación CU 16	2	08/07/2017	
10/07/2017	Testing CU 16	4	13/07/2017	
14/07/2017	Testing Integral 7	2	15/07/2017	Fernandez y Arizaga

Duración estimada en días con un tiempo de desarrollo de 12 hs semanales por cada miembro del equipo.

En la 1° entrega el equipo se compromete a presentar los casos de uso 20, 17, 18, 3, 2, 5, 1, 4, 12 y 13 testeados y documentados.

En la 2° entrega el equipo se comprometa a presentar los casos de uso 9, 6, 15, 7, 14, 19, 10, 8, 11 y 16 testeados y documentados.

7- Sembranza de los candidatos

Base de Datos:

1. Microsoft SQL Server: Confección del DER con herramienta case Studio. Creación de stored procedure con manejo de transacciones y manejo de errores, inserción, modificación y eliminación de datos, consultas complejas relaciones de tablas, confección de sub consultas.
2. MySQL: confección del DER, conversión del mismo a BD, creación de rutinas básicas Creación de stored procedure y funciones; inserción, modificación y eliminación de datos, consultas complejas relaciones de tablas. Use de herramientas MySQL workbench y PHPMyAdmin.

Programación:

1. C# .Net: manejo de Herencia, colecciones, conexión con motor de base de datos SQLServer y MySQL, manejo de la colección de objetos, creación de controles dinámicos, implementación de interfaces, clases abstractas
2. ASP .Net: manejo de Herencia, colecciones, conexión con motor de base de datos SQLServer y MySQL, validaciones Java Script, utilización de masterpage, implementación de interfaces, clases abstractas

3. Java: creación de clases, manejo de herencia, implementación de interfaces, manejo de Servlets, conexión con MySQL.
4. PHP: conexión con Base de datos MySQL Validación JavaScript y PHP, inclusión de archivos, creación e implementación de clases, codificación y decodificación a JSON, Impresión en PDF, generación de gráficos con chart.js