



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**APLICACIÓN WEB PARA REALIZAR LA GESTIÓN DISTRIBUIDA DE GASTOS DE
UN HOGAR**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

JAIME SEBASTIÁN SANZ BIAVA

PROFESOR GUÍA:

SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:

PABLO BARCELÓ BAEZA

MARÍA CECILIA RIVARA ZÚÑIGA

SANTIAGO DE CHILE

2016

Resumen

El trabajo desarrollado en esta memoria nace producto de la necesidad de llevar a cabo un registro transparente de los gastos de una vivienda compartida por estudiantes. Actualmente las universidades tienen muchos estudiantes de regiones, que vienen a ciudades más grandes (por ejemplo, a Santiago) en busca de un título profesional. Esos estudiantes típicamente comparten casa o departamento con otros estudiantes, como una forma de reducir el costo de estudiar lejos de sus casas.

Actualmente estos estudiantes cuentan con diversos métodos manuales para llevar a cabo, tanto el registro como balance de los gastos de la vivienda que comparten. Sin embargo, dichos registros suelen no ser muy transparentes, donde además se suelen perder los respaldos (por ejemplo, las boletas o comprobantes) de los gastos incurridos. Por lo tanto, la administración de gastos de este tipo de viviendas es no sólo engorrosa, sino también una fuente frecuente de problemas entre las personas que comparten el mismo techo.

Para ayudar a paliar esta situación, este trabajo de memoria propone una solución tecnológica que resuelve los problemas de transparencia y balance, dejando un registro permanente de los gastos realizados. La solución desarrollada cuenta con un módulo servidor (Web), el cual permite a los usuarios ingresar gastos incluyendo una foto de las boletas o tickets de respaldo. Además, permite la creación de listas de compras, para facilitar así el abastecimiento de los víveres para el hogar. Permite también crear presupuestos, los cuales ayudan a organizar los gastos.

Por último, la aplicación cuenta con un módulo que permite realizar el balance de gastos. Para ello, muestra gráficamente las contribuciones de cada usuario, y genera instrucciones de traspaso de fondos para que cada persona contribuya con un monto similar al final de cada mes.

Tanto durante como al finalizar este desarrollo, se contó con retroalimentación de usuarios reales que ayudaron a mejorar la interfaz y los servicios provistos por la aplicación. Particularmente, al final del ciclo de desarrollo se realizó una marcha blanca, que permitió realizar un diagnóstico y evaluación de la implementación realizada. Los resultados obtenidos fueron bastante buenos, aunque se identificaron algunos elementos de mejora, especialmente en la usabilidad de las interfaces.

Por último, y luego de analizar el *feedback* obtenido, se propusieron nuevas metas para extender este trabajo en el futuro.

Para mis papás, Jaime Sanz y Mónica Biava, por haberme dado los medios para emprender este viaje, y el apoyo necesario para terminarlo.

Agradecimientos

Este trabajo corresponde al fruto de un proceso de aprendizaje a lo largo del cual he tenido la suerte de contar con mucho apoyo, tanto en lo académico como en lo personal, por lo que son muchas las personas a las que debo agradecer.

Antes que nada, agradecerle a mi madrina, Francisca Sanz, por haberme apoyado en mis primeros años lejos de mis tierras. A mi hermano Nicolás, por la compañía (y sobre todo la música) en estos últimos 5 años viviendo juntos, así como también a los *roommates* que hemos tenido en este tiempo, Sven Reisenegger y Martín Mege. Gracias por haber llenado de historias el departamento.

Le agradezco a mis amigos de la sección 3 por todos esos *tacas* mágicos que nunca olvidaré, y por haberme dado tantos buenos recuerdos.

También quiero agradecer a quienes han estado conmigo en estos últimos años haciéndome compañía y dándome su apoyo y cariño: Matías Yañez, Daniel Aviv, Fernando Morales, Elizabeth Marchant, Natalia Hernández, Cesar Arriagada, Oscar Guajardo, Bernardo Küpfer, Tomás Wolf, Sofía Valenzuela, Javiera Galvéz, y tantos otros que se me quedan en el tintero.

Le agradezco también a César por haber sido el usuario más activo durante la marcha blanca, y uno de los mayores proveedores de *feedback*. Le agradezco también a mi hermana Francisca y a mi hermano Diego, así como a mis papás, por haberse dado el tiempo de probar la plataforma y haberme dado sus opiniones a lo largo del desarrollo.

Una mención especial para Daniel y Bernardo, por haber sido siempre mis compañeros en tantos proyectos y tareas; gracias por mantenerme motivado, y por mostrarme que las metas pueden cumplirse, por muy difícil que parezca.

Por último, quiero agradecer a mi profesor guía, Sergio Ochoa, por toda la ayuda prestada tanto en el desarrollo de la aplicación como en la redacción de este documento. Gracias por creer en el proyecto.

Tabla de Contenido

1.	Introducción.....	1
1.1.	Problema abordado	1
1.2.	Justificación de la propuesta.....	2
1.3.	Objetivos de la memoria	2
1.4.	Contenidos del documento de memoria	3
2.	Análisis de las soluciones existentes	4
2.1.	Expense.....	4
2.2.	Tricount.....	4
2.3.	CoinKeeper.....	5
2.4.	PayMeKaty	5
2.5.	iXpenselt	5
2.6.	LiveExpenses	5
2.7.	iSpending	6
2.8.	MoneyControl.....	6
2.9.	Soluciones ad hoc.....	6
2.10.	Comparación de funcionalidades.....	7
3.	Requisitos de la solución	9
3.1.	Requisitos de usuario	9
3.2.	Requisitos de software.....	10
4.	Diseño de la solución	14
4.1.	Arquitectura física de la solución	14
4.2.	Arquitectura lógica de la solución	15
4.2.1.	Subsistema de login	16
4.2.2.	Subsistema de administración de datos de usuario.....	16
4.2.3.	Subsistema de administración de usuarios de una vivienda	17
4.2.4.	Subsistema de administración de invitaciones.....	17
4.2.5.	Subsistema de administración de categorías	18
4.2.6.	Subsistema de administración de gastos	18
4.2.7.	Subsistema de administración de ítems.....	19

4.2.8.	Subsistema de administración de listas de compras	20
4.2.9.	Subsistema de administración de presupuestos	21
4.2.10.	Subsistema de administración de vacaciones	22
4.2.11.	Subsistema de balanceo de gastos.....	23
4.3.	Mapa de navegación	24
4.3.1.	Módulos de administración de usuarios	26
4.3.2.	Módulos de administración de balance	26
4.3.3.	Módulos de administración de gastos	26
4.3.4.	Módulos de administración de presupuestos	27
4.3.5.	Módulos de administración de listas.....	28
4.3.6.	Módulos de administración de ítems.....	28
4.3.7.	Módulos de administración de categorías	28
4.3.8.	Módulos de administración de vacaciones	29
4.4.	Modelo de datos	29
5.	Implementación de la solución	34
5.1.	Sobre el template base y los CSS utilizados.....	34
5.2.	Registro y login.....	35
5.2.1.	Registro	35
5.2.2.	Recuperación de contraseña.....	37
5.2.3.	Login.....	39
5.3.	Cómo formar parte de una vivienda	40
5.3.1.	Crear una nueva vivienda.....	40
5.3.2.	Unirse a vivienda existente	41
5.4.	Sobre la página de configuración de la vivienda	43
5.5.	Invitar usuarios.....	44
5.6.	Manejo de categorías.....	45
5.6.1.	Categorías por <i>default</i>	46
5.6.2.	Crear categoría.....	47
5.6.3.	Categorías especiales	49
5.6.4.	Tipos de categorías	50
5.6.5.	Eliminar categoría	50

5.6.6.	Modificar Categoría.....	51
5.7.	Manejo de gastos.....	53
5.7.1.	Sobre el proceso de confirmación de gastos.....	53
5.7.2.	Crear gasto de forma directa	54
5.7.3.	Crear gasto a partir de lista de compras	61
5.7.4.	Confirmar gasto.....	62
5.7.5.	Modificar gasto	63
5.7.6.	Eliminar gasto.....	65
5.7.7.	Graficar gastos	66
5.7.8.	Exportar XLS	67
5.7.9.	Sobre la privacidad.....	68
5.8.	Manejo de presupuestos	70
5.8.1.	Listado de presupuestos	70
5.8.2.	Crear presupuesto.....	70
5.8.3.	Crear desde presupuesto existente	71
5.8.4.	Modificar.....	73
5.8.5.	Gráficos	73
5.9.	Manejo de ítems	74
5.9.1.	Listado.....	74
5.9.2.	Crear	75
5.9.3.	Editar.....	76
5.10.	Manejo de lista de compras	77
5.10.1.	Crear lista de compras	77
5.10.2.	Modificar lista de compras.....	80
5.11.	Completar una lista de compras	82
5.12.	Manejo de vacaciones.....	87
5.12.1.	Crear vacación.....	88
5.12.2.	Modificar vacación	89
5.13.	Sobre el balance.....	91
5.13.1.	Visualizar balance.....	91
5.13.2.	Sobre el cálculo de los balances.....	93

5.14.	Manejo de transferencias	95
6.	Evaluación de la solución	98
6.1.	Sobre la interfaz y los primeros pasos.....	99
6.2.	Gastos y categorías	99
6.3.	Listas de compras e ítems	100
6.4.	Sobre la reutilización de presupuestos	100
6.5.	Sobre el balance y las transferencias	101
6.6.	Otros comentarios	101
7.	Conclusiones y trabajo futuro	102
8.	Bibliografía	103
Anexo A.	Resultados de la encuesta web	104

Índice de Tablas

Tabla 1. Comparación de funcionalidades de las herramientas analizadas.....	7
Tabla 2. Matriz de trazado de requisitos.....	13
Tabla 3. Tabla de totales para los usuarios del ejemplo 1.....	93
Tabla 4. Tabla de totales para los usuarios del ejemplo 2.....	94
Tabla 5. Tabla de totales para los usuarios del ejemplo 3.....	95

Índice de Figuras

Figura 1. Arquitectura física de la solución.....	14
Figura 2. Arquitectura lógica de Django.....	15
Figura 3. Subsistema de login.....	16
Figura 4. Subsistema de administración de datos de usuario.....	16
Figura 5. Subsistema de administración de usuarios de una vivienda.....	17
Figura 6. Subsistema de administración de invitaciones.....	17
Figura 7. Subsistema de administración de categorías.....	18
Figura 8. Subsistema de administración de gastos.....	19
Figura 9. Subsistema de administración de ítems.....	20
Figura 10. Subsistema de administración de listas de compras.....	21
Figura 11. Subsistema de administración de presupuestos.....	22
Figura 12. Subsistema administrador de vacaciones.....	23
Figura 13. Subsistema de balanceo de gastos.....	24
Figura 14. Mapa de navegación para usuario no autenticado.....	24
Figura 15. Mapa de navegación para usuario sin vivienda activa.....	25
Figura 16. Diagrama general del mapa de navegación para un usuario con vivienda activa.....	25
Figura 17. Módulos de administración de usuarios.....	26
Figura 18. Módulos de administración de balance de gastos.....	26
Figura 19. Módulos de administración de gastos.....	27
Figura 20. Módulos de administración de presupuestos.....	27
Figura 21. Módulos de administración de listas.....	28
Figura 22. Módulos de administración de ítems.....	28
Figura 23. Módulos de administración de categorías.....	29
Figura 24. Módulos de administración de vacaciones.....	29
Figura 25. Modelo Entidad-Relación de la base de datos.....	32
Figura 26. Entidades de Django para la administración de usuarios. Esta figura extiende el modelo presentado en la Figura 25 a través del modelo auth_user.....	33
Figura 27. Barra de navegación para un usuario que no se ha autenticado.....	34
Figura 28. Barra de navegación para usuario autenticado, pero sin vivienda activa.....	34
Figura 29. Barra de navegación para un usuario autenticado con vivienda activa.....	34
Figura 30. Página de registro.....	36
Figura 31. Formulario impide repetir nombre de usuario y/o email.....	36

Figura 32. Se envía mail de activación.....	36
Figura 33. Mail de activación con link para activar.	37
Figura 34. Cuenta activada.	37
Figura 35. Link para recuperar contraseña.....	37
Figura 36. Formulario para recuperar contraseña.	38
Figura 37. Se envía mail con link para reestablecer contraseña.	38
Figura 38. Correo con link de para reestablecer contraseña.....	38
Figura 39. Formulario para reestablecer contraseña.	38
Figura 40. Contraseña reestablecida exitosamente.	39
Figura 41. Formulario de ingreso a la plataforma.	39
Figura 42. Página de inicio para usuario sin vivienda activa.	39
Figura 43. Página de inicio para usuario con vivienda activa. Corresponde a la página de configuración de la vivienda.	40
Figura 44. Formulario para crear una nueva vivienda.....	40
Figura 45. Luego de crear la vivienda, se redirige al usuario a la página de configuración de la nueva vivienda.....	41
Figura 46. Lista de invitaciones, donde se ve una invitación de parte del usuario <i>Allie</i> para formar parte de la vivienda <i>El Depa</i>	42
Figura 47. Detalle invitación recibida.	42
Figura 48. Al aceptar la invitación, se redirige a la configuración de la vivienda.....	43
Figura 49. Página de configuración de la vivienda.	43
Figura 50. Mensaje de ayuda de la página de configuración.	44
Figura 51. Formulario para invitar usuarios a la vivienda.....	45
Figura 52. Lista de invitaciones enviadas.	45
Figura 53. Usuario puede cancelar invitación enviada.....	45
Figura 54. Listado de categorías por <i>default</i> de la vivienda.	46
Figura 55. Formulario para crear nueva categoría.	47
Figura 56. Mensaje de alerta al intentar crear categoría personalizada con nombre “Agua”. La validación se hace en el lado del servidor, por lo es necesario enviar el formulario para obtener el mensaje.	48
Figura 57. Usuario ingresa nombre de categoría personalizada.	48
Figura 58. El usuario es redirigido al listado de categorías luego de ingresar categoría personalizada. La categoría ingresada puede verse en el listado.	49
Figura 59. Se muestra ícono para eliminar categoría. No es posible eliminar categorías default. ...	51
Figura 60. La categoría fue eliminada. Notar que la categoría “Personal_1” no aparece en la tabla, pues esta debiese encontrarse entre “Servicios” y “Otros”, como se vio en la Figura 59.	51
Figura 61. Formulario de edición de categoría.....	52
Figura 62. Se editan los atributos de la categoría.	52
Figura 63. Se editan los campos de la categoría “Agua”, y los cambios se ven reflejados en el listado de categorías.....	52
Figura 64. Combinación inválida de atributos de la categoría.	53
Figura 65. Mensaje de error luego de introducir combinación inválida de atributos de la categoría.53	
Figura 66. Proceso de confirmación de gastos. Las flechas en negrita representan un cambio de	

estado del gasto, el cual es gatillado por la acción indicada: pagar, confirmar o editar. Notar que el usuario que inicia el proceso y el que paga el gasto pendiente no son necesariamente el mismo. 54

Figura 67. Formulario para crear nuevo gasto. Notar que se tiene seleccionada la opción *Pendiente*. 55

Figura 68. Se pide confirmación del usuario antes de crear el gasto. 55

Figura 69. Nuevo gasto pendiente..... 56

Figura 70. Detalle de gasto pendiente..... 56

Figura 71. Detalle de gasto con estado *por confirmar*. El campo *Usuario* indica al usuario que pagó el gasto..... 57

Figura 72. Formulario para nuevo gasto pagado..... 57

Figura 73. *Datepicker* para facilitar el ingreso de fechas. La fecha en verde claro (3) indica la fecha seleccionada, mientras que la amarilla (8) indica el día actual. 58

Figura 74. Datos para generar nuevo gasto pagado..... 59

Figura 75. Listado de gastos por confirmar. Puede verse también el gasto de la Figura 69. 59

Figura 76. Detalle de gasto con estado *por confirmar* con foto de boleta..... 60

Figura 77. *Izquierda*: Foto de la boleta del gasto de la Figura 76 luego del *resize*. La foto fue tomada con un smartphone *moto G 2015*, y tenía una resolución de 720x1280 pixeles antes del *resize* hecho por el servidor. *Derecha*: Para enfatizar que el tamaño de la imagen es suficiente, se muestra una foto de una página cualquiera de la edición de bolsillo del libro *It* de Stephen King. La foto fue tomada con el mismo smarthpone. La fuente del libro resulta considerablemente más pequeña que la de la mayoría de las boletas emitidas. 61

Figura 78. Detalle de gasto con estado *por confirmar*, desde el punto de vista de un usuario que no pagó el gasto..... 62

Figura 79. Gasto aparece como *pagado* luego de ser confirmado por todos los usuarios de la vivienda. Detalle del gasto desde el punto de vista de un usuario que no pagó. 63

Figura 80. Detalle de gasto pagado por el mismo usuario. 64

Figura 81. Vista para editar una instancia de gasto pagado..... 64

Figura 82. Se modifica el monto del gasto, lo cual a su vez revierte el estado de *pagado a por confirmar*. 65

Figura 83. Luego de eliminar el gasto, se redirige al usuario a la lista de gastos. 66

Figura 84. Interfaz para generar los gráficos del historial de gastos..... 66

Figura 85. Gráficos del total de gastos por categoría y por período. 67

Figura 86. Descarga del historial de gastos en formato XLS..... 68

Figura 87. Archivo XLS con el historial de gastos..... 68

Figura 88. El detalle de los gastos se obtiene mediante el método GET..... 69

Figura 89. El usuario es redireccionado a la página de errores al intentar ingresar a un gasto externo..... 69

Figura 90. Listado de presupuestos. 70

Figura 91. Formulario para crear nuevo presupuesto..... 71

Figura 92. Se crea nuevo presupuesto para el período 2016-6..... 71

Figura 93. Formulario para reutilizar los datos de un presupuesto existente para crear uno nuevo. 72

Figura 94. Al cargar los datos, el formulario de fondo cambia sus valores por los del presupuesto para *Gas* del período 2016-6. 72

Figura 95. Formulario para modificar los valores de un presupuesto dado.....	73
Figura 96. Interfaz que muestra gráficamente los presupuestos definidos para un período dado. Al colocar el puntero sobre las distintas partes del gráfico puede verse el detalle.	73
Figura 97. Listado de ítems de la vivienda.	75
Figura 98. Formulario para crear nuevo ítem.....	75
Figura 99. Nuevo ítem aparece en el listado de ítems de la vivienda. Notar el uso del filtro de la tabla para facilitar encontrar el ítem deseado.	76
Figura 100. Formulario de edición de ítem.	76
Figura 101. Se edita el ítem y pueden verse los nuevos valores.	77
Figura 102. Formulario inicial para crear una nueva lista.	77
Figura 103. Formulario para lista con 2 ítems.	78
Figura 104. Select2 permite filtrar el listado de ítems.	79
Figura 105. Al seleccionar un ítem, se le indica al usuario la unidad de medida apropiada, en este caso “botella”.	79
Figura 106. El usuario debe confirmar los datos antes de crear la lista.	80
Figura 107. Formulario para completar lista.	81
Figura 108. Formulario para editar una lista de compras.	81
Figura 109. El usuario intenta modificar la lista con campos inválidos, por lo que la lista no se guarda, y se le muestran los errores.	82
Figura 110. Formulario para completar lista de compras visto desde un smartphone <i>moto G 2015</i> .83	
Figura 111. El usuario puede crear una nueva lista con los ítems que no se compraron en esta instancia.....	84
Figura 112. <i>Izquierda</i> : Se pide confirmación del usuario antes de marcar los ítems indicados como comprados. <i>Derecha</i> : se redirige al usuario al detalle del gasto asociado a la lista pagada.	85
Figura 113. Detalle de la lista de compras pagada.	86
Figura 114. Lista de compras creada a partir de los ítems que no fueron comprados en una lista anterior.	87
Figura 115. Formulario para crear nueva instancia de vacación.	88
Figura 116. Se utiliza jquery-ui para facilitar el correcto ingreso de las fechas.	88
Figura 117. Se valida que la fecha final sea posterior a la inicial.....	89
Figura 118. Listado de vacaciones definidas para los usuarios de la vivienda, donde puede verse la vacación que se acaba de crear.....	89
Figura 119. Formulario de edición de vacaciones.	90
Figura 120. Se cambia la fecha de fin de la vacación.	90
Figura 121. Los nuevos valores pueden verse en el listado de vacaciones.	91
Figura 122. Representación gráfica del balance de gastos.....	92
Figura 123. Instrucciones para balancear.....	92
Figura 124. Formulario para realizar transferencias visto desde el balance de gastos.....	96
Figura 125. Formulario de transferencias.....	96
Figura 126. Usuario selecciona un monto y usuario para transferir. Notar que la transferencia corresponde a la que sugirió el sistema en la Figura 123.....	96
Figura 127. Transferencias son visibles en el listado de gastos por confirmar.	97
Figura 128. Gasto generado automáticamente para <i>Allie</i>	97

Figura 129. Gasto generado automáticamente para <i>Levni</i>	97
Figura 130. Una vez que se confirman las transferencias apropiadas, el balance de gastos indica que los usuarios se encuentran en equilibrio.....	98

1. Introducción

En Chile se da el caso de que muchos estudiantes de regiones optan por estudiar en la Región Metropolitana en lugar de sus ciudades de origen. La Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile no es una excepción, teniendo cerca de un 27% de alumnos procedentes de regiones [1]. Muchos de estos estudiantes, e inclusive jóvenes que se van a vivir de manera independiente por razones ajenas a la educación, suelen compartir una vivienda mientras estudian o trabajan. Desde una perspectiva económica cada una estas personas opera en forma independiente, pero usualmente aportan dinero para alimentación básica, pago de servicios, gastos comunes y/o arreglos del lugar donde viven. Estos aportes individuales suelen quedar respaldados en algún papel (tickets, boletas, etc.), el cual es susceptible a perderse si la persona no es cuidadosa con las cuentas. Esto genera a su vez problemas de convivencia y hace difícil mantener balanceados los aportes de la gente.

Por otra parte, resulta difícil llevar un registro del estado de la vivienda; es decir, qué tan abastecida está, por lo que esta información no suele estar inmediatamente disponible al momento de realizar una compra de mercadería. Finalmente, al llevar un balance informal sobre los gastos y contribuciones, es difícil contar con información histórica que les permita a los individuos organizarse y prever sus gastos. Por ejemplo, es sabido que durante el invierno suben los gastos de energía eléctrica y/o gas, debido a la calefacción.

Dada la diversidad de aspectos a considerar para llevar la cuenta de los gastos y las contribuciones de estas personas, resulta difícil poder operar con un único mecanismo, por lo que usualmente se emplean varios, cada uno de los cuales sirve a un propósito específico. Sin embargo, el uso de diversos mecanismos resulta tedioso y poco práctico. Este trabajo de memoria busca desarrollar una aplicación web que permita a las personas que conviven en una vivienda llevar los gastos y contribuciones de manera fácil, ordenada y segura.

1.1. Problema abordado

Se consideró principalmente el problema de visibilidad y gestión distribuida y segura de los datos asociados a la administración de un hogar. Los datos manejados son tanto los de carácter monetario, como aquellos que tienen relación con el estado de la vivienda. Este problema se aborda mediante el desarrollo de una aplicación web enfocada a resolver y evitar los problemas descritos en la sección 1. Dicha aplicación es capaz de llevar un registro de los gastos de cada individuo, así como una lista de los víveres y suplementos faltantes, la cual es accesible por los usuarios, los cuales pueden consultarla y actualizarla en tiempo real y de forma remota, por ejemplo, desde sus *smartphones*.

De esta forma, se resuelven los problemas de coordinación que se generan cuando 2 integrantes no se ponen de acuerdo para realizar compras, ya que se les está entregando un canal de

comunicación de fácil acceso, el cual cuenta con información actualizada sobre el nivel de abastecimiento de la vivienda.

Por otro lado, al tener los gastos de cada integrante en una base de datos estructurada no sólo se facilita llevar a cabo el balance de las cuentas y gastos de cada uno, sino que además se hace posible desarrollar estadísticas sobre los distintos gastos que se realizan: gastos por individuo, gastos por cuenta, gastos mensuales, etc. Esta información resulta sumamente útil, ya que puede utilizarse para organizar los gastos, tanto de la vivienda como de cada individuo.

1.2. Justificación de la propuesta

Más allá de los desafíos propios de desarrollar una aplicación de esta naturaleza, este trabajo de memoria tiene importantes desafíos de usabilidad, seguridad y trazabilidad de la información. Respecto a la usabilidad, la aplicación debe resultar fácil de usar, puesto que en caso contrario los potenciales usuarios optarían por seguir utilizando los mecanismos tradicionales de papel y lápiz, desechando así cualquier posibilidad de mejora que pudiera entregarle esta aplicación. El proceso de administración de dicha información puede hacerse de la manera más rápida y simple posible, minimizando así el riesgo a que algún usuario no utilice efectivamente la aplicación.

Por otro lado, la aplicación trabaja con datos sumamente sensibles, por lo que ésta debe garantizar la seguridad de los servicios y la confiabilidad y trazabilidad de la información. De esta forma, los usuarios no se sienten vulnerados al utilizarla. Se implementa también un mecanismo para la validación de los fondos ingresados, de tal forma que ningún usuario malicioso sea capaz de falsificar la información compartida.

La aplicación desarrollada es además suficientemente liviana y robusta, y corre de forma fluida en las plataformas más utilizadas por los usuarios, ya sea en un navegador móvil o de escritorio.

Finalmente, la validación del producto final resulta una tarea no trivial, pues es necesario validar utilizando usuarios reales que entreguen retroalimentación respecto a la usabilidad y utilidad de la aplicación. De esta manera es posible comparar sus experiencias con y sin la aplicación, para lo cual resulta necesario definir parámetros de evaluación que sean suficientemente representativos para el caso.

1.3. Objetivos de la memoria

El objetivo general de este trabajo de memoria consiste en diseñar e implementar la aplicación web, que esta sea capaz de correr en múltiples plataformas de hardware, y que permita administrar en forma distribuida los gastos y recursos de un hogar, particularmente en casas o departamentos compartidos por estudiantes. La aplicación debe ser fácil de utilizar, segura y debe permitir la trazabilidad de la información compartida y de las operaciones que sobre ésta se ejecutan. Para lograr este objetivo general, se propusieron los siguientes objetivos específicos:

- Diseñar e implementar un repositorio de gastos y recursos compartidos, el cual puede ser administrado en forma distribuida.
- Diseñar e implementar un mecanismo de trazabilidad de datos y operaciones sobre estos.
- Diseñar e implementar mecanismos de seguridad que minimicen la vulnerabilidad de los datos y servicios administrados por la aplicación.
- Integrar estos componentes en una aplicación web capaz de ejecutar en plataformas de hardware que van desde una laptop hasta un teléfono inteligente.

1.4. Contenidos del documento de memoria

Este documento de memoria consta de 8 capítulos. El capítulo 2 revisa y discute las soluciones existentes para administrar gastos tanto personales como compartidos. En el capítulo 3 se definen los requisitos de usuario y software que debe cumplir la solución. El capítulo 4 propone un diseño de solución cuyo objetivo consiste en cumplir los requisitos previamente definidos. En el capítulo 5 se muestran todas las interfaces de la implementación actual de la solución, y se explica como el usuario interactúa con cada una de ellas. En el capítulo 6 se intenta hacer una evaluación de la implementación, y se discuten los puntos fuertes y débiles de ésta. El capítulo 7 muestra las conclusiones del proyecto, y se discuten las mejoras que se pueden construir sobre la base implementada. El capítulo 8 contiene la bibliografía utilizada.

2. Análisis de las soluciones existentes

Para identificar las soluciones más frecuentemente utilizadas, se entrevistó a diversos estudiantes de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, quienes comparten alojamiento con otros estudiantes, y por lo tanto se ven en la necesidad de llevar los gastos en forma distribuida. Además, se realizó una encuesta web a través de la plataforma *U-Cursos*, tanto a nivel facultad como Universidad¹. Por último, se buscaron en Internet más alternativas, las cuales se describen a continuación.

2.1. Expense

Expense consiste en una aplicación móvil para iOS cuyo objetivo es ayudar a llevar registro de los gastos personales que se realizan. La aplicación permite tener varias “carteras”, a las cuales se le pueden destinar usos distintos, por ejemplo, cartera para gastos personales y para gastos de la vivienda. El usuario puede hacer depósitos a las carteras, para luego generar gastos, los cuales llevan un título y fecha, y caen en alguna categoría predefinida. Los gastos también tienen un campo opcional para adjuntar una foto, por ejemplo, de una boleta.

De esta forma, la aplicación es capaz de generar gráficos que indican los gastos por mes, año o semestre. Estos gráficos además permiten visualizar en qué grado cada categoría contribuye al gasto total del período. Finalmente, es posible generar presupuestos mensuales para cada categoría, para más tarde analizar si estos se excedieron, o si fueron sobre-estimados.

Si bien la aplicación permite generar más de una cartera, éstas sólo son accesibles para el usuario que las crea, y están sujetas al perfil del usuario en la *AppStore*, por lo que no es posible compartir carteras, ni ver los gráficos de otros usuarios. Es decir, no es posible generar “grupos”. Además, no existe una versión web, y sólo está disponible para iOS. Por último, no es posible generar categorías personalizadas, y los gastos no llevan detalle, sólo un título.

2.2. Tricount

Tricount consiste en una aplicación multiplataforma² que permite balancear los gastos en un grupo. Es posible definir varios grupos distintos, a los cuales se agregan varios alias para identificar a cada integrante. Luego, dentro del grupo se pueden ir agregando gastos, los cuales consisten de un título, una breve descripción, un monto, quién pagó el gasto y finalmente los miembros del grupo que deben contribuir al gasto (no son necesariamente todos).

¹ Los resultados se adjuntan en el anexo A

² iOS, Android y Web

De esta forma, se van acumulando gastos, los cuales son pagados por los distintos integrantes. Se permite además generar una URL especial para poder ingresar al grupo desde cualquier plataforma, y poder acceder a todas las funcionalidades sin necesidad de registrarse (la información de las versiones móviles se sincroniza con la de la URL generada). La aplicación genera gráficos donde se ven las contribuciones de cada usuario (balances positivos/negativos).

Finalmente, cuando el grupo lo considera apropiado, se pueden balancear los gastos, lo que genera una lista de instrucciones que indican cuánto y a quien le debe pagar cada usuario (de la forma "X debe pagar Y a Z"), de forma que todos los balances queden iguales.

Si bien el balance de los gastos funciona muy bien, es el ingreso de estos el que podría generar conflictos, ya que no se permite incluir muchos detalles, ni tampoco incluir ningún tipo de documento que respalde la veracidad de los montos (como podría ser el ingreso de una boleta).

2.3. CoinKeeper

Aplicación para iOS y Android similar a Expense. En lugar de tratar a los ingresos como instancias particulares de gastos (que es lo que ocurría en Expense), estos se tratan de manera distinta, lo que permite generar gráficos mensuales y anuales de gastos vs ingresos. Además, introduce el concepto de múltiples fuentes de dinero, lo que permite generar informes sobre ingresos y gastos por fuente, lo que puede resultar útil cuando se tienen, por ejemplo, múltiples cuentas bancarias. La interfaz resulta muy usable, al hacer uso de *drag-and-drop* para gatillar la creación de un gasto. Por último, permite la creación de categorías personalizadas y de presupuestos para éstas.

2.4. PayMeKaty

Aplicación para iOS muy similar a Tricount, con la diferencia que se exige que los usuarios de los grupos estén registrados en la aplicación para poder tener acceso a los balances. Tiene el beneficio adicional de poder, al momento de ingresar un gasto al grupo, asignarles ponderaciones distintas a los usuarios. Además, genera algunas estadísticas (gastos por mes y año).

2.5. iXpenselt

Aplicación iOS, similar a Expense y CoinKeeper. Tiene una gran jerarquía de subcategorías, además de permitir categorías personalizadas. También permite enviar tablas de gastos e ingresos al correo, en distintos formatos (PDF, HTML o CSV). Tiene variedad de visualizaciones (gráficos de barras, tortas e informes).

2.6. LiveExpenses

Aplicación iOS similar a Expense. No agrega funcionalidades nuevas: sólo permite generar informes sobre gastos e ingresos y enviarlos al mail. No requiere registro.

2.7. iSpending

Aplicación iOS sumamente usable y *user-friendly*. Parecida a Expense y CoinKeeper. Si bien no existe la diferenciación de carteras que usa Expense, es posible agregar categoría a los ingresos para indicar su procedencia (salario, inversión, negocio, etc.). Permite visualizar gráficos de ingresos vs gastos por mes, con el beneficio adicional de generar una ventana de tiempo a partir de éste, la cual puede ser de 3, 6 o 12 meses. No hay categorías personalizadas, pero se tienen los gráficos de gastos por categoría. Por último, surge un nuevo tipo de gráfico: porcentajes de procedencia de los ingresos. No requiere registro.

2.8. MoneyControl

Aplicación multiplataforma (iOS, Android, MacOS y Windows). Tiene la funcionalidad adicional de programar ingresos/gastos que se realizan de forma periódica. También hace uso de la categorización de los ingresos.

2.9. Soluciones ad hoc

Las antes descritas son algunas de las soluciones informáticas disponibles para resolver el problema de la administración de gastos e ingresos. Sin embargo, la mayoría están enfocadas a gastos personales, y ninguna lleva el registro del estado de la vivienda.

En general este problema no se toma en cuenta (al menos en los casos de los estudiantes con quienes se habló sobre el tema), y los pocos que intentaban resolverlo utilizaban hojas de cálculo de Google Drive³. De esta manera se cuenta con un documento accesible de forma concurrente por los miembros de la vivienda, el cual contiene información básica (lista de víveres que faltan, por ejemplo). Además, el documento se puede acceder fácilmente desde el *Smartphone* usando la aplicación de Google Drive.

El problema de esta solución está en la usabilidad del documento, ya que, al ser simplemente una hoja de cálculo con texto, no es fácil *marcar* un ítem de la lista como comprado, por lo que se suele borrar la fila correspondiente. Esto a su vez genera otro problema de usabilidad: no se pueden reutilizar los ítems de la lista, y cada vez que se quiera agregar algo es necesario escribir el nombre completo nuevamente. Lo anterior provoca que la solución rápidamente caiga en desuso.

Claramente, este instrumento no fue diseñado para cumplir este rol, por lo que no es muy criticable el hecho de que no tenga soporte para estos servicios.

³ <https://www.google.com/intl/es/drive/>

2.10. Comparación de funcionalidades

En la siguiente tabla se muestra un resumen de las funcionalidades de las aplicaciones mencionadas anteriormente.

Tabla 1. Comparación de funcionalidades de las herramientas analizadas.

Funcionalidad / Aplicación	Expense	Tricount	CoinKeeper	PayMeKaty	iXpenseIt	LiveExpenses	iSpending	MoneyControl
Datos compartidos		X		X				
Categorías Personalizadas		X	X	X	X			
Carteras o Categorías para Ingresos	X		X			X	X	X
Gastos por Categoría	X		X		X	X	X	X
Ingresos por Categoría			X			X	X	X
Gastos vs Ingresos			X		X	X	X	X
Entradas periódicas								X
Variedad de Gráficos					X			
Balanceo de gastos		X		X				
Registro Obligatorio	X			X				
Realizar Presupuestos	X		X					
Adjuntar Foto (Boleta)	X							
Envío de Informes vía email					X	X		

Tomando en cuenta los datos entregados en la tabla, sumado a la experiencia como usuario y considerando además el contexto del problema que se pretende resolver, se propone rescatar las estadísticas generadas por las aplicaciones, y ajustarlas para que puedan ser utilizadas para llevar los gastos compartidos de la vivienda, en lugar de los gastos personales.

Además, se busca que la aplicación permita realizar el balance de los gastos, y la modificación de forma concurrente de las listas de compras de la vivienda. Además, se propone rescatar el concepto de “cartera” visto anteriormente para identificar a los miembros de la vivienda, de tal forma que cada usuario sea capaz de ver estadísticas del fondo común.

En resumen, se propone que la aplicación cuente al menos con las siguientes características y funcionalidades:

- *Multiplataforma*: Resulta necesario que la aplicación sea multiplataforma, ya que el éxito de ésta se basa principalmente en que todos los miembros de la vivienda puedan utilizarla.
- *Liviana y sencilla*: La interfaz debe ser lo suficientemente amigable y sencilla para que no sea necesario un tutorial sobre cómo ingresar datos. *iSpending* y *CoinKeeper* resaltan entre las otras aplicaciones por la sencillez del diseño y la rapidez con que el usuario se acostumbra a utilizarlas.
- *Soporte para información compartida*: los datos y gráficos deben ser accesibles por todos los miembros de la vivienda.
- *Soporte para categorías personalizadas*: se considera necesario ya que el no encontrar las categorías deseadas puede fomentar el abandono de la aplicación.
- *Soporte para el ingreso de gastos*: para fomentar la sencillez y facilidad de uso de la aplicación, se propone que sólo se ingresen gastos, dejando de lado el concepto de ingreso.
- *Entrega de estadísticas con ventanas de tiempo variables*: es decir, gastos por categoría por período.
- *Balanceo de Gastos (con ponderación)*: pieza fundamental para mantener equilibradas las contribuciones de cada usuario.
- *Creación de presupuestos*: se considera pertinente agregarlo, pues resulta útil ya que ayuda a organizar los gastos a medida que pasa el mes.
- *Ítems*: corresponden a los víveres y suplementos necesarios en la vivienda. Los usuarios deben ser capaces de generar ítems personalizados de acuerdo a sus propias necesidades. Estos ítems facilitan la creación de listas de compras.
- *Listas de Compras*: Los usuarios son capaces de crear listas de compras *online* las cuales son accesibles por cualquier usuario de la Vivienda. Gracias a la lista de ítems definida por lo usuarios, resulta posible reutilizar los ítems ingresados en listas anteriores, facilitando así la creación de nuevas listas de compras, superando así los problemas que tenía la solución del *Google Doc*.

3. Requisitos de la solución

A continuación, se describen los principales requisitos de usuario y de software:

3.1. Requisitos de usuario

Tomando en cuenta las funcionalidades listadas anteriormente, se definen los siguientes requisitos de usuario:

- RU001 El sistema debe autenticar usuarios, y sólo así permitir el acceso a los datos y servicios del mismo.
- RU002 El sistema debe implementar un mantenedor de viviendas.
- RU003 Mantenedor de usuarios de una vivienda.
- RU004 Mantenedor de categorías de gastos.
- RU005 Mantenedor de gastos.
- RU006 Los usuarios deben tener acceso a una visualización sencilla que muestre el balance de gastos por usuario.
- RU007 Se debe poder adjuntar una imagen al momento de crear un nuevo gasto.
- RU008 El sistema debe permitir generar una lista de instrucciones indicando cómo equilibrar los gastos entre los miembros de la vivienda.
- RU009 El sistema debe permitir la realización de transferencias de fondos (ficticios) entre los usuarios, para indicarle así al sistema que los gastos se encuentran equilibrados.
- RU010 Mantenedor de presupuestos por período por categoría.
- RU011 Los usuarios deben tener acceso a visualizaciones acerca de los gastos por categoría por período.
- RU012 Mantenedor de ítems de la vivienda.
- RU013 Mantenedor de listas de compras.
- RU014 Los usuarios deben poder completar una lista de compras.
- RU015 Los usuarios deben poder generar períodos de ausencia o de vacaciones durante los cuales no se encontrarán en la Vivienda.
- RU016 El sistema debe permitir la generación del historial de gastos en formato XLS.

- RU017 La aplicación debe ser usable en múltiples dispositivos.

3.2. Requisitos de software

Se proponen los siguientes requisitos de software con el fin de cumplir los requisitos de usuario:

- RS001 Crear, modificar y eliminar cuenta de usuario.
- RS002 Ingresar al sitio usando la cuenta de usuario.
- RS003 Recuperar la contraseña vía email.
- RS004 Crear, modificar y eliminar vivienda asociada al usuario activo.
- RS005 Invitar usuarios a una vivienda:
 - se notifica a los usuarios vía email para aceptar/declinar la invitación.
 - se muestra una notificación en la aplicación, donde también se da la opción de aceptar/declinar.
- RS006 Ingreso de gasto:
 - Se debe indicar monto y categoría del gasto.
 - El gasto puede ingresarse como *pendiente* o como *pagado*.
 - En caso de marcarlo como *pagado*, se da la opción de especificar la fecha de pago, siendo el *default* el día actual.
- RS007 Los gastos deben pasar por 3 estados:
 - Pendiente: el gasto aún no se paga.
 - Por confirmar: el gasto fue pagado por un usuario, pero debe ser confirmado por todos los usuarios activos de la vivienda antes de marcarlo como pagado.
 - Pagado: el gasto fue pagado por un usuario, y ya fue confirmado por todos los usuarios activos de la Vivienda.
- RS008 Modificación de gasto:
 - Si se encuentra pendiente, puede ser modificado por cualquier usuario activo
 - Si se encuentra por confirmar o pagado:
 - sólo puede ser modificado por el usuario que pagó.

- Luego de la modificación, el estado queda como por confirmar.
- RS009 Un gasto pendiente puede ser eliminado por cualquier usuario.
- RS010 Un gasto por confirmar o pagado sólo puede ser eliminado por el usuario que pagó.
- RS011 El gráfico de balances de gastos debe indicar:
 - Usuarios con balance positivo, es decir, usuarios que han pagado más de lo que deberían.
 - Usuarios con balance negativo: es decir, usuarios que han pagado menos de lo que deberían.
 - La suma de los balances debe ser 0.
- RS012 El cálculo de balances debe tomar en cuenta sólo categorías marcadas como compartidas.
- RS013 El cálculo de balances debe considerar períodos de ausencia de los usuarios.
- RS014 Ingreso de foto asociada a gasto:
 - puede tener formato PNG o JPG. Tamaño mínimo 250x250 px, y máximo 500x500 px. Si se ingresa una foto de mayor tamaño, se hace un *resize* para ajustarse a estos límites.
- RS015 Generación de instrucciones para balance de gastos:
 - Se calculan los gastos de cada individuo, y se genera una lista de instrucciones del estilo “[usuario1] debe pagar [monto] a [usuario2]” las cuales apuntan a dejar iguales los gastos de cada usuario.
- RS016 Al hacer una transferencia se deben generar 2 nuevos gastos:
 - Uno con monto positivo a nombre del usuario que transfiere.
 - Uno con monto negativo a nombre del usuario al que se le transfiere.
 - Los montos deben ser iguales en valor absoluto.
- RS017 Las transferencias deben poder diferenciarse del resto de los gastos.
- RS018 Las transferencias se generan con estado por confirmar, y deben ser confirmadas por todos los usuarios activos de la vivienda antes de pasar al estado pagado.
- RS019 Las transferencias no pueden ser modificadas ni eliminadas por los usuarios.

- RS020 Crear, modificar y eliminar presupuesto por categoría.
- RS021 Visualización de gráfico de gastos por categoría.
- RS022 Crear, modificar y eliminar ítems para la vivienda.
- RS023 Crear, modificar y eliminar lista de compras.
 - Las listas de compras se crean utilizando los ítems almacenados en la base de datos, en lugar de texto plano
- RS024 Al completar y pagar una lista, se debe crear un gasto de categoría “Supermercado” por el monto indicado al pagar la lista. El gasto debe estar a nombre del usuario que completó la lista.
- RS025 Al pagar una lista, se debe permitir omitir ítems de la lista, y dar la opción de:
 - Descartar los ítems no comprados
 - Crear una nueva lista a partir de los ítems no comprados
- RS026 Crear, modificar y eliminar períodos de ausencia.
- RS027 Generación de documento XLS con historial de gastos.
- RS028 La aplicación debe ser usable en dispositivos iOS.
 - Se deben definir las reglas de estilo necesarias.
- RS029 La aplicación debe ser usable en dispositivos Android.
 - Se deben definir las reglas de estilo necesarias.
- RS030 La aplicación debe ser usable en Firefox y Chrome.
 - Se deben definir las reglas de estilo necesarias.

A continuación, se presenta la matriz de trazado de los requisitos.

Tabla 2. Matriz de trazado de requisitos.

	RU 01	RU 02	RU 03	RU 04	RU 05	RU 06	RU 07	RU 08	RU 09	RU 10	RU 11	RU 12	RU 13	RU 14	RU 15	RU 16	RU 17
RS01	X																
RS02	X																
RS03	X																
RS04		X															
RS05			X														
RS06				X	X												
RS07					X												
RS08					X												
RS09					X												
RS10					X												
RS11						X											
RS12						X											
RS13						X											
RS14							X										
RS15								X									
RS16									X								
RS17									X								
RS18									X								
RS19									X								
RS20										X							
RS21											X						
RS22												X					
RS23													X				
RS24														X			
RS25														X			
RS26															X		
RS27																X	
RS28																	X
RS29																	X
RS30																	X

4. Diseño de la solución

A continuación, se describe la arquitectura física y lógica de la solución, y el modelo de datos del sistema.

4.1. Arquitectura física de la solución

La arquitectura física corresponde a una estructura convencional para soportar sistemas web (tres capas), tal como se muestra en la Figura 1.

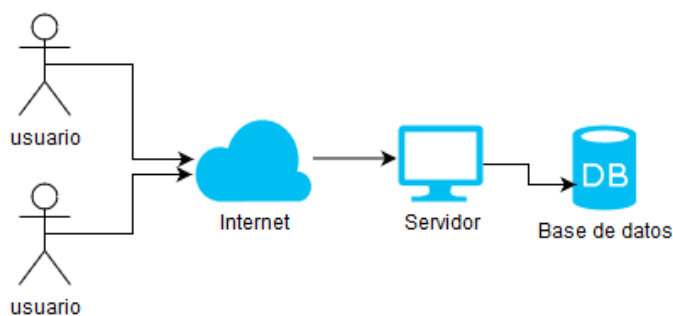


Figura 1. Arquitectura física de la solución.

Para acceder a los servicios, los usuarios envían peticiones al servidor a través de un browser. El servidor a su vez responde a las solicitudes, y hace las consultas pertinentes a la base de datos, la cual almacena la información del sistema y los usuarios.

La arquitectura física fue implementada utilizando el *framework* Django⁴ en su versión estable más reciente⁵, en conjunto con Python 3.4.3⁶ y PostgreSQL⁷ para las bases de datos. Se eligieron estas tecnologías por sobre otras alternativas (por ejemplo, *Ruby on Rails*⁸ o *CakePHP*⁹) debido, principalmente, a que ya se cuenta con experiencia trabajando con éstas.

⁴ <https://www.djangoproject.com/>

⁵ Versión 1.9.5 (al 19 de mayo de 2016)

⁶ <https://www.python.org/downloads/release/python-343/>

⁷ <http://www.postgresql.org/>

⁸ <http://rubyonrails.org/>

⁹ <http://cakephp.org/>

4.2. Arquitectura lógica de la solución

Las aplicaciones de Django se construyen siguiendo una variación del *Modelo Vista Controlador*, según la lógica presentada en la Figura 2.

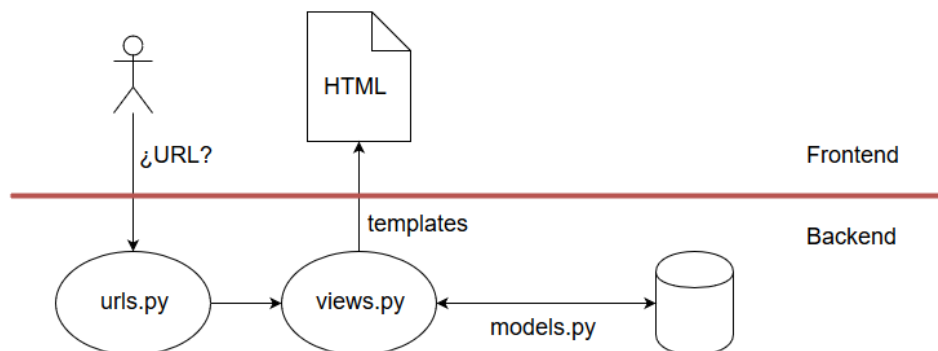


Figura 2. Arquitectura lógica de Django.

El proceso es el siguiente:

- El usuario solicita una URL desde su navegador.
- En el lado del servidor, Django procesa la URL solicitada usando el módulo `urls.py`, el cual le delega la solicitud a una vista específica, la cual se encuentra definida en `views.py`.
- Las vistas definidas en `views.py` hacen las *queries* pertinentes a la base de datos a través de los modelos definidos en `models.py`.
- Luego de obtener los datos solicitados a la base de datos, éstos son procesados de manera pertinente. El archivo HTML que finalmente se le entrega al usuario es generado a partir de la composición de:
 - o Los datos procesados por la vista
 - o Un archivo HTML definido previamente (al cual se le llama *template*), el cual presupone la existencia de los datos entregados por la vista, e indica cómo estos deben ser presentados al usuario.

El módulo `models.py` contiene las definiciones de todas las tablas de la base de datos, las cuales a su vez tienen asociados métodos que facilitan la generación de solicitudes a ésta por medio del *Object-relational Mapper (ORM)*¹⁰ de Django, el cual permite abstraerse de la sintaxis de SQL.

¹⁰ <https://www.fullstackpython.com/object-relational-mappers-orms.html>

Estos métodos, además, son capaces de hacer procesamientos posteriores luego de obtener las respuestas a las solicitudes, de forma tal de facilitar el trabajo realizado por la vista. Se puede pensar en estos modelos como los *administradores* de las entidades del modelo de datos.

De esta forma, la arquitectura lógica del sistema puede descomponerse en varios subsistemas, los cuales se detallan a continuación.

4.2.1. Subsistema de login

Este subsistema (Figura 3) es el responsable de responder a las solicitudes de entrada a la aplicación. El módulo procesador de consultas de identificación se encarga de responder de manera adecuada las solicitudes.

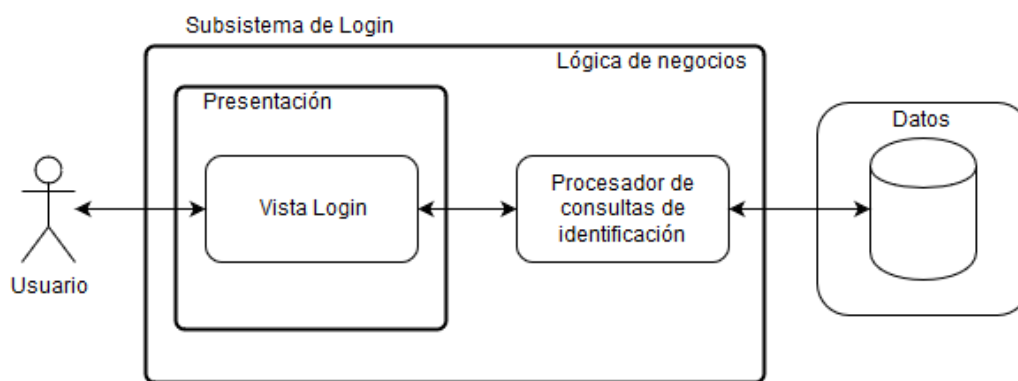


Figura 3. Subsistema de login.

4.2.2. Subsistema de administración de datos de usuario

Se encarga de administrar los datos de la cuenta que el usuario tiene guardado (Figura 4). Tiene un único módulo, el cual se encarga de la modificación de los datos: email, alias y contraseña.

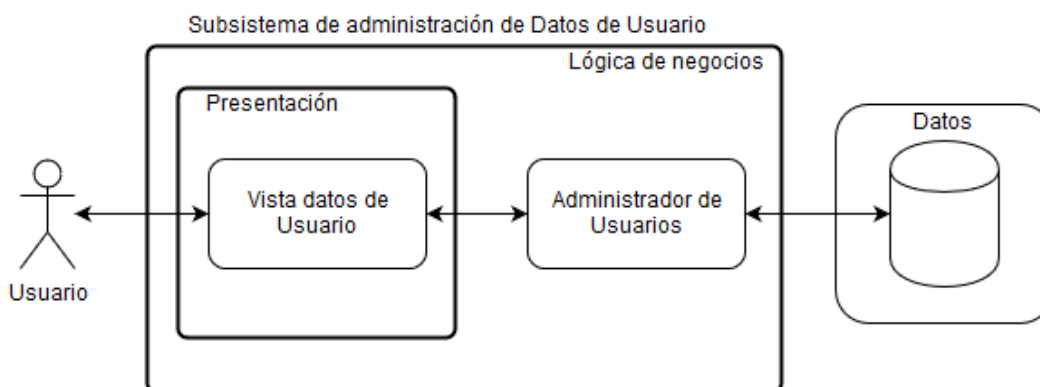


Figura 4. Subsistema de administración de datos de usuario.

4.2.3. Subsistema de administración de usuarios de una vivienda

Responsable de la administración de los usuarios pertenecientes a la vivienda (Figura 5). El módulo de administración de vivienda se encarga de la creación y la vista de la vivienda, mientras que el módulo de administrador de invitaciones es el encargado de enviar invitaciones a otros usuarios. Además, el módulo de administración de vivienda se encarga del abandono de la vivienda por parte del usuario activo.

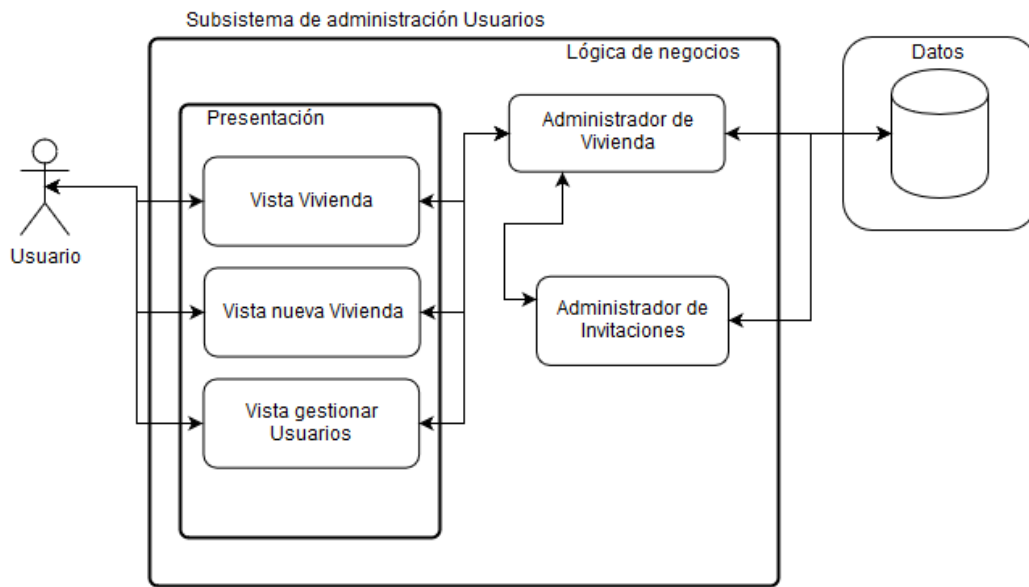


Figura 5. Subsistema de administración de usuarios de una vivienda.

4.2.4. Subsistema de administración de invitaciones

Este subsistema (Figura 6) se encarga de administrar las respuestas a las invitaciones a viviendas, pudiendo cancelar o aceptar estas. Contiene un único módulo, el administrador de Invitaciones, el cual se encarga de las respuestas.

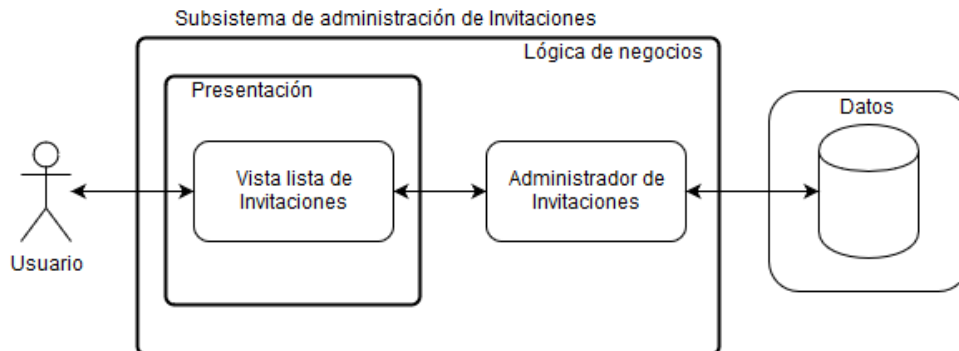


Figura 6. Subsistema de administración de invitaciones.

4.2.5. Subsistema de administración de categorías

Encargado de la creación, modificación y eliminación de categorías (Figura 7). Se encarga también de la creación de las categorías por default creadas al crear una nueva vivienda.

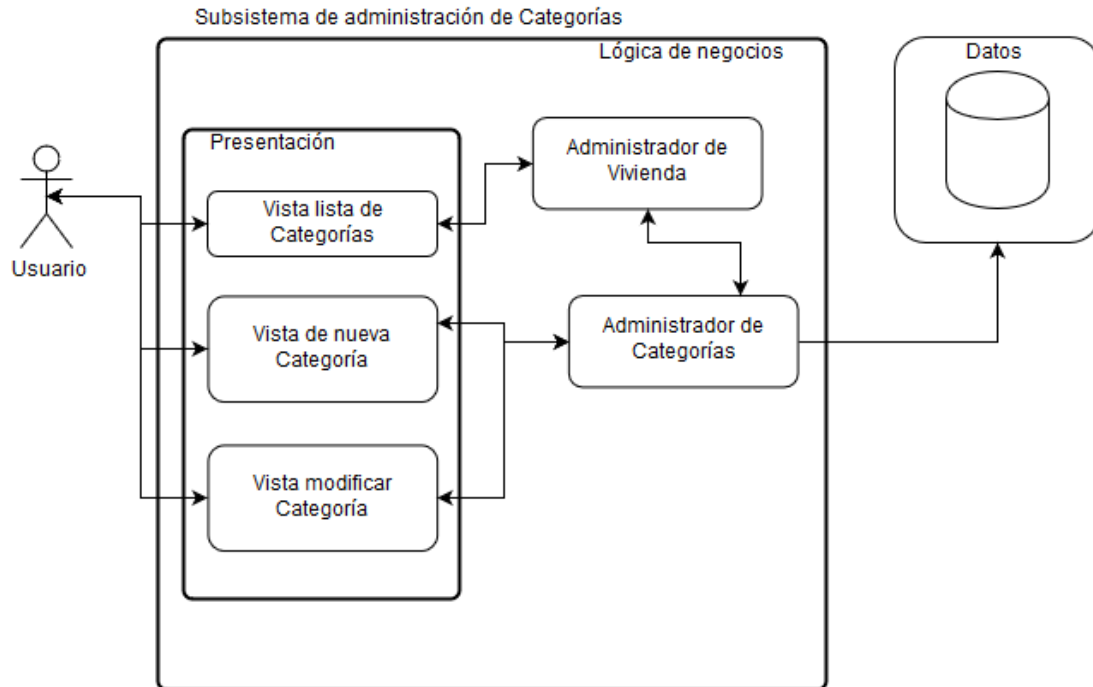


Figura 7. Subsistema de administración de categorías.

4.2.6. Subsistema de administración de gastos

El administrador de vivienda delega estas responsabilidades al módulo administrador de gastos, el cual se encarga de la creación y modificación de gastos, así como de la generación del historial de gastos en formato XLS. El ingreso de foto es responsabilidad del administrador de fotos. Este proceso puede verse en la Figura 8.

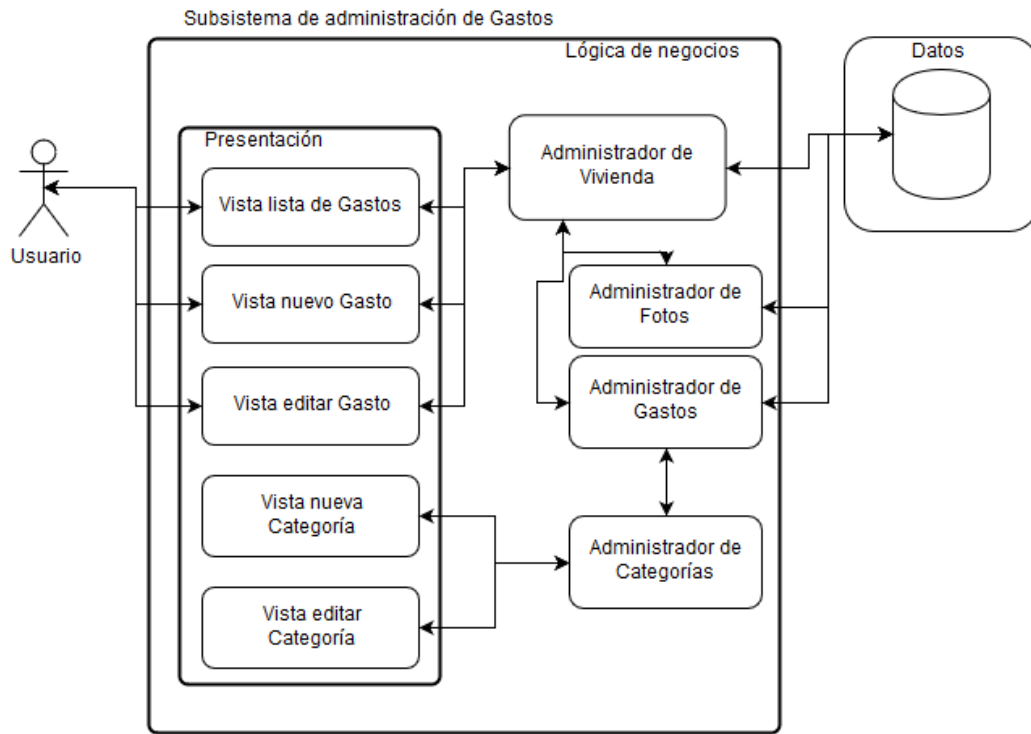


Figura 8. Subsistema de administración de gastos.

4.2.7. Subsistema de administración de ítems

Se encarga de la creación y modificación los ítems de la vivienda, además de la creación de los ítems por default que se crean al crear una nueva vivienda (Figura 9).

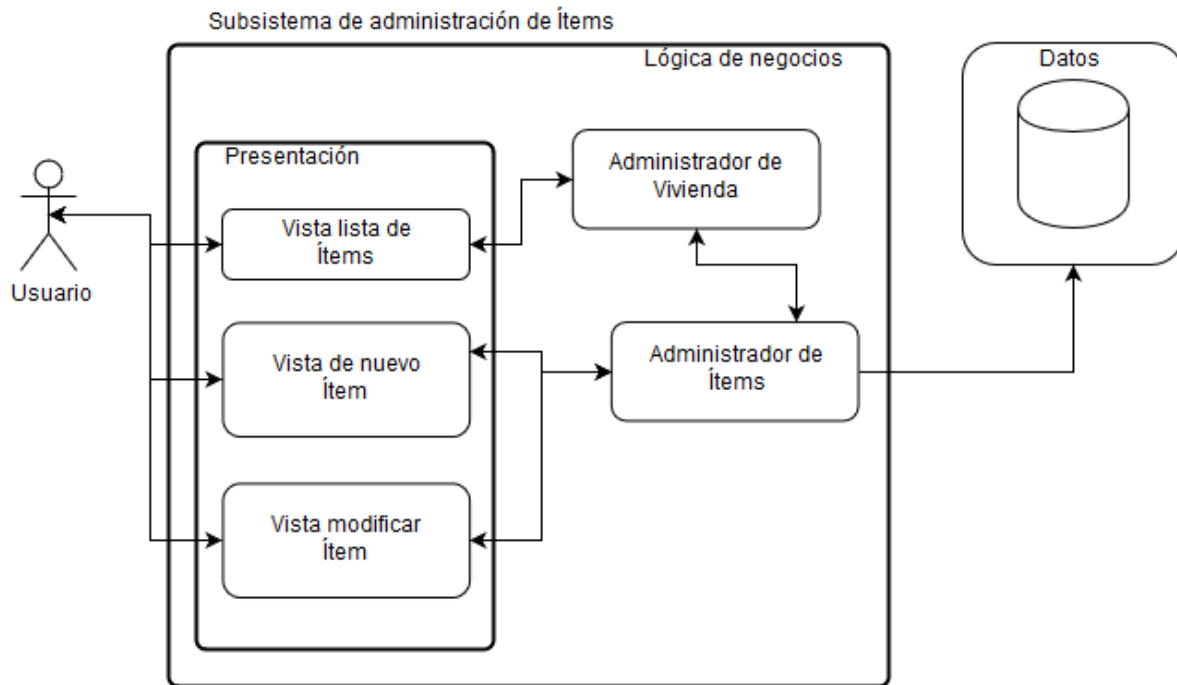


Figura 9. Subsistema de administración de ítems.

4.2.8. Subsistema de administración de listas de compras

Encargado de la creación, edición y posterior pago¹¹ de listas de compras asociadas a la vivienda (Figura 10). Al momento de pagar una lista de compras, se genera una nueva instancia de gasto, por lo que el administrador de listas le delega esta responsabilidad al administrador de gastos y al administrador de fotos.

¹¹ Se refiere al proceso de *completar* una lista, es decir, marcar los ítems de la lista como *comprados*, para luego generar un gasto asociado a la lista.

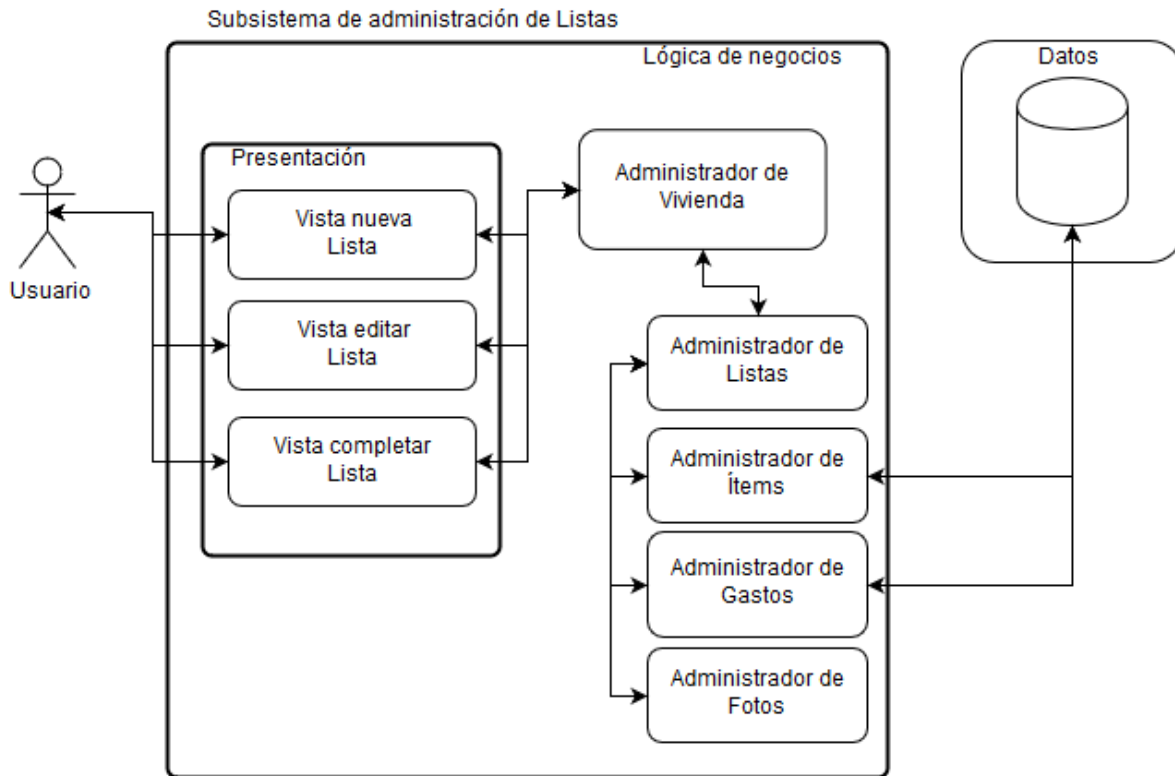


Figura 10. Subsistema de administración de listas de compras.

4.2.9. Subsistema de administración de presupuestos

Encargado de generar y modificar presupuesto para la vivienda (Figura 11). El administrador de vivienda delega estas responsabilidades al administrador de presupuestos. El administrador de gastos permite computar el total invertido en cada categoría, de tal forma de mostrar los porcentajes de completitud de cada presupuesto hasta el momento.

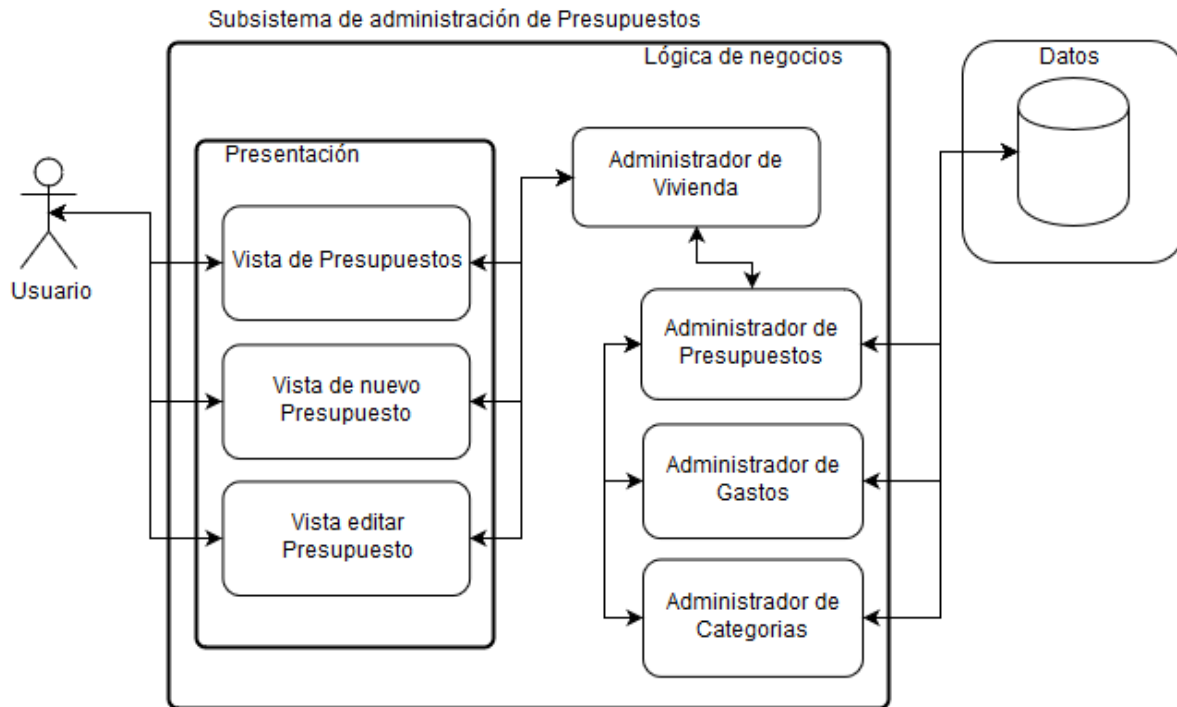


Figura 11. Subsistema de administración de presupuestos.

4.2.10. Subsistema de administración de vacaciones

El administrador de vivienda le delega las siguientes responsabilidades: generación, edición y eliminación de vacaciones/salidas de los usuarios de la vivienda (Figura 12).

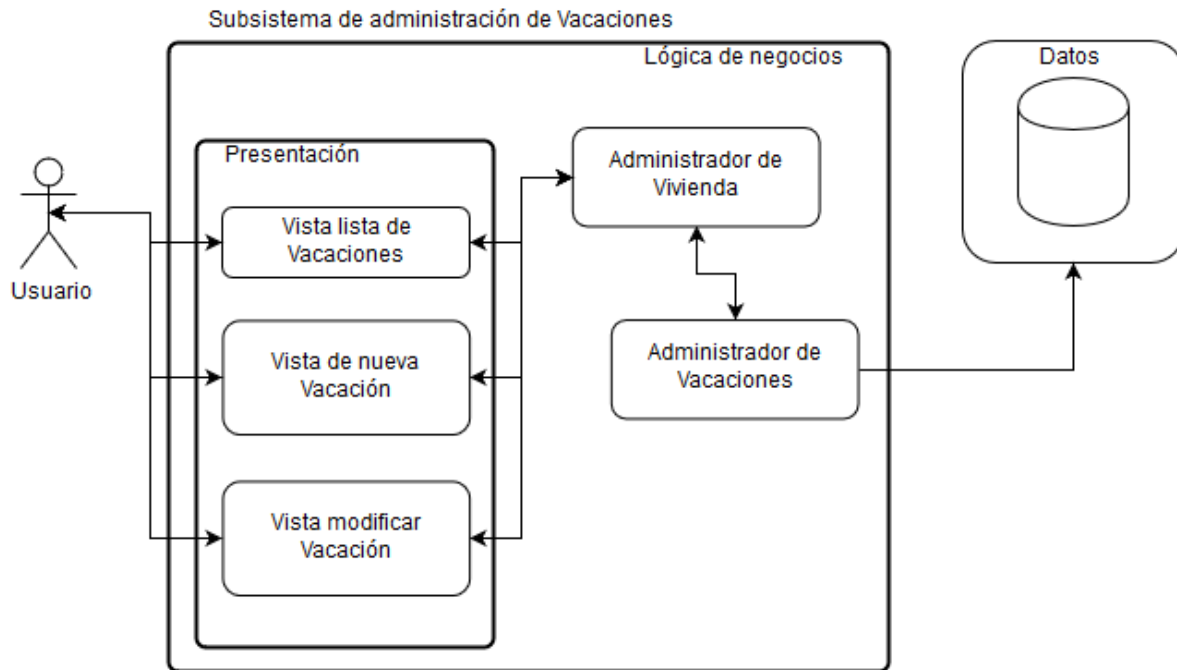


Figura 12. Subsistema administrador de vacaciones.

4.2.11. Subsistema de balanceo de gastos

Responsable de generar instrucciones para que los usuarios puedan llevar a cabo el ajuste de cuentas (Figura 13), quedando todos con la misma contribución al gasto común de la vivienda. El módulo administrador de gastos, en este sistema, aporta información sobre los gastos de cada usuario, mientras que el módulo administrador de categorías aporta información sobre las categorías que deben considerarse al momento de computar los totales de cada usuario.

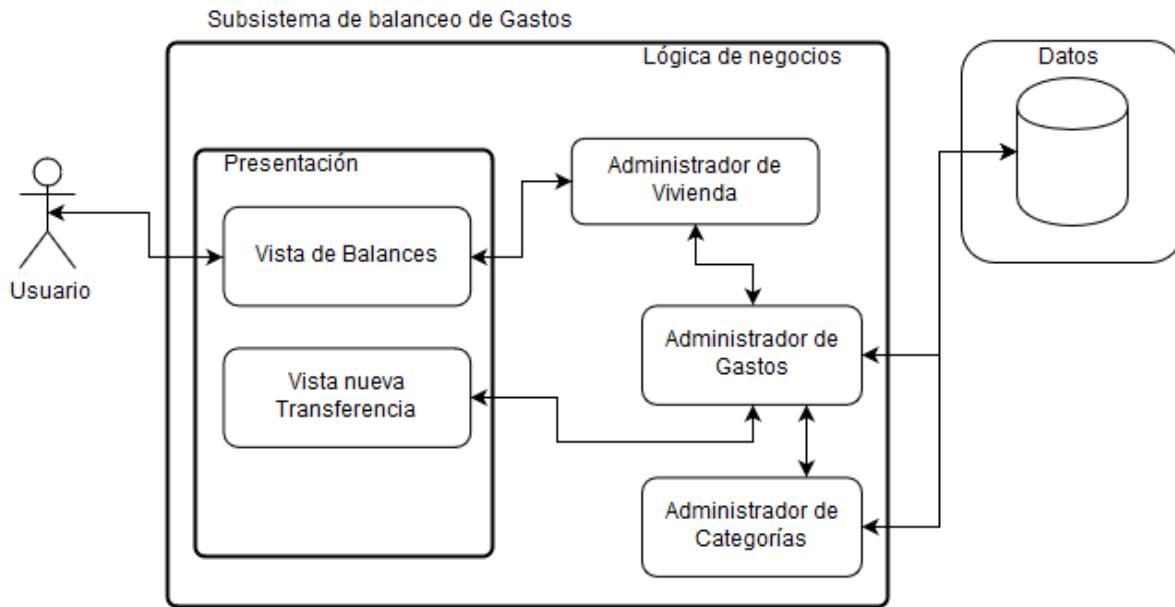


Figura 13. Subsistema de balanceo de gastos.

4.3. Mapa de navegación

La Figura 14 muestra el mapa de navegación para un usuario no autenticado. Una vez que el usuario ingresa al sitio, y sólo para efectos de esta sección, se consideran 2 tipos de usuario:

- Usuario sin vivienda.
- Usuario con vivienda.

La Figura 15 muestra el mapa de navegación para un usuario sin vivienda, mientras que la Figura 16 muestra el mapa de navegación para un usuario con vivienda. Debido a la complejidad de este último, el detalle de los módulos más importantes se describe por separado en las subsecciones siguientes.

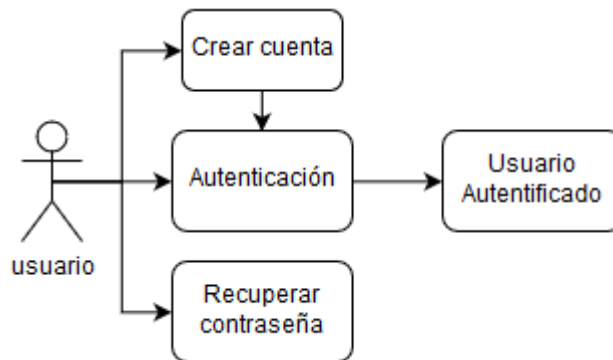


Figura 14. Mapa de navegación para usuario no autenticado.

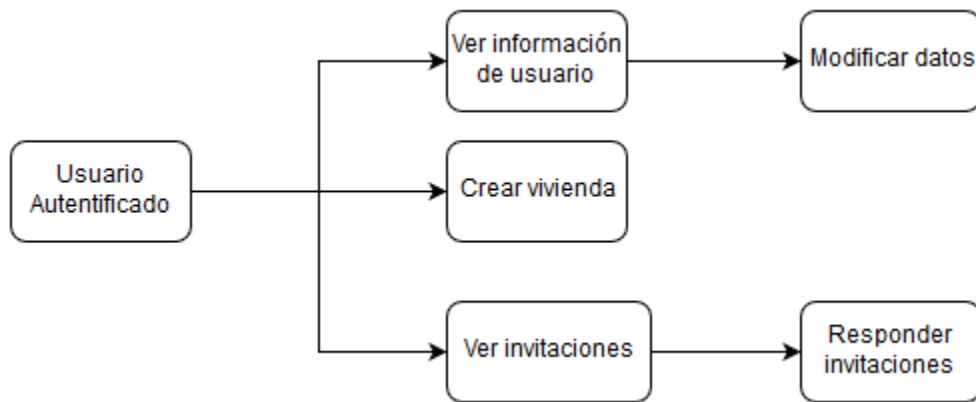


Figura 15. Mapa de navegación para usuario sin vivienda activa.

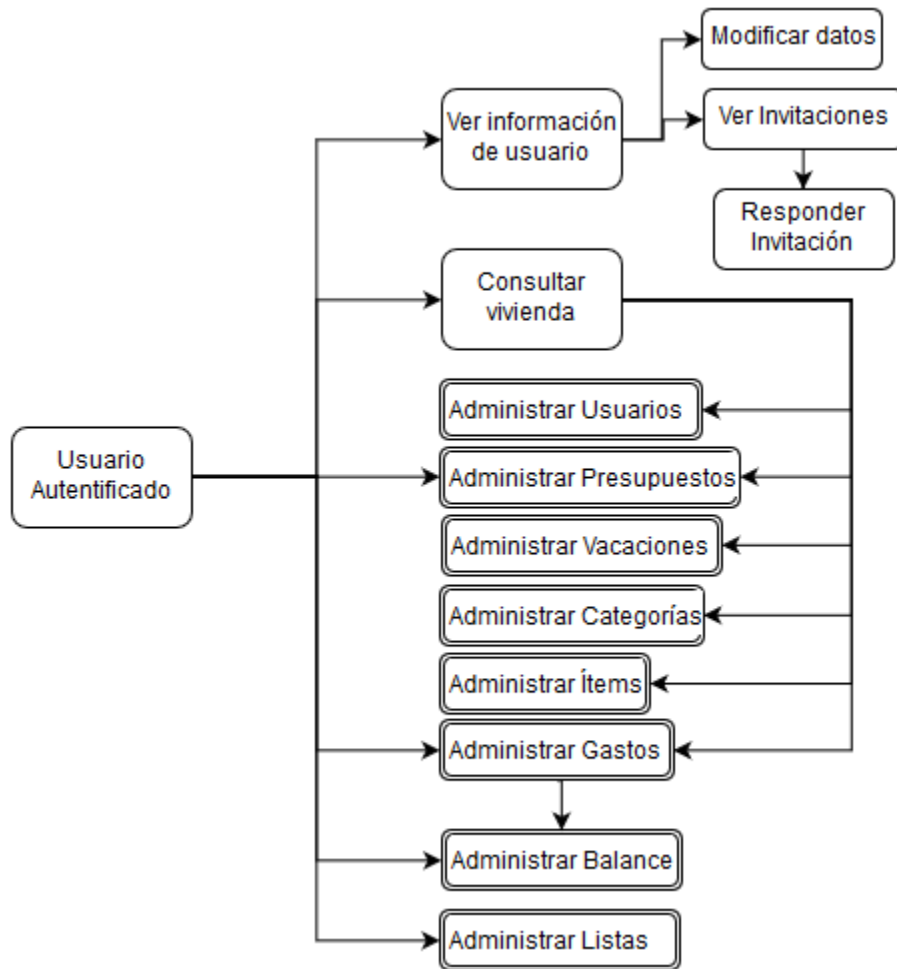


Figura 16. Diagrama general del mapa de navegación para un usuario con vivienda activa.

En las siguientes subsecciones se muestra el detalle de la funcionalidad considerada para los principales módulos administradores que son parte de la solución.

4.3.1. Módulos de administración de usuarios

Se encargan de la generación de invitaciones de nuevos usuarios a la vivienda y de la realización de transferencias entre usuarios. Además, se encargan de la acción de abandonar la vivienda activa (Figura 17).

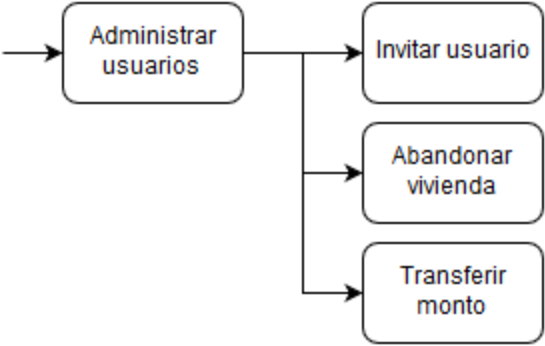


Figura 17. Módulos de administración de usuarios.

4.3.2. Módulos de administración de balance

Se encargan de las acciones referentes al balanceo de los gastos de los miembros de una vivienda (Figura 18). Específicamente, de mostrar gráficamente el balance de gastos entre los usuarios y de mostrar las instrucciones para que los usuarios puedan llevar a cabo el ajuste de cuentas.

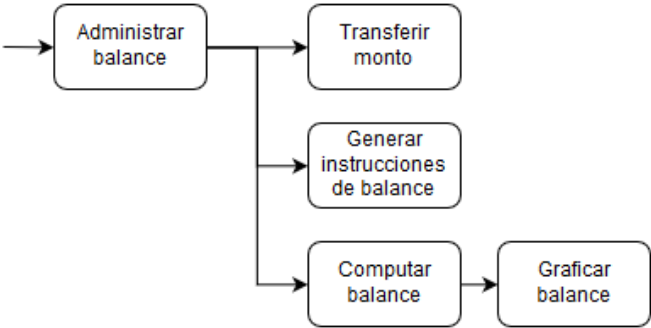


Figura 18. Módulos de administración de balance de gastos.

4.3.3. Módulos de administración de gastos

Encargados de la administración de los gastos (Figura 19); permiten ingresar nuevos gastos, y modificar otros ya existentes. Del mismo modo, son los encargados del ingreso de fotos. Por

último, permiten la generación y posterior descarga en formato XLS del historial de gastos de la vivienda.

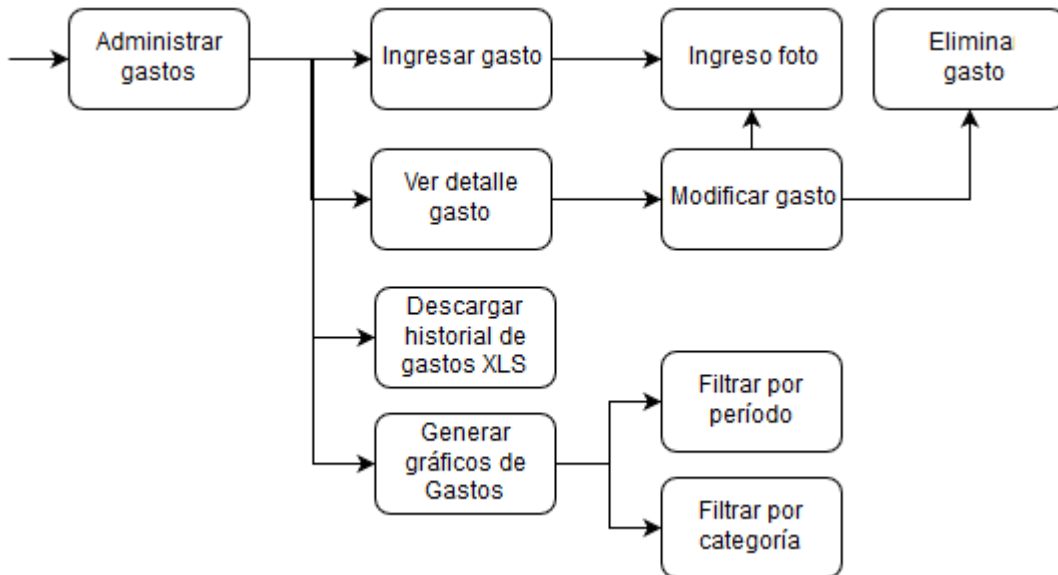


Figura 19. Módulos de administración de gastos.

4.3.4. Módulos de administración de presupuestos

Se encargan de crear, modificar y mostrar los presupuestos definidos para los distintos períodos y categorías, como puede verse en la Figura 20.

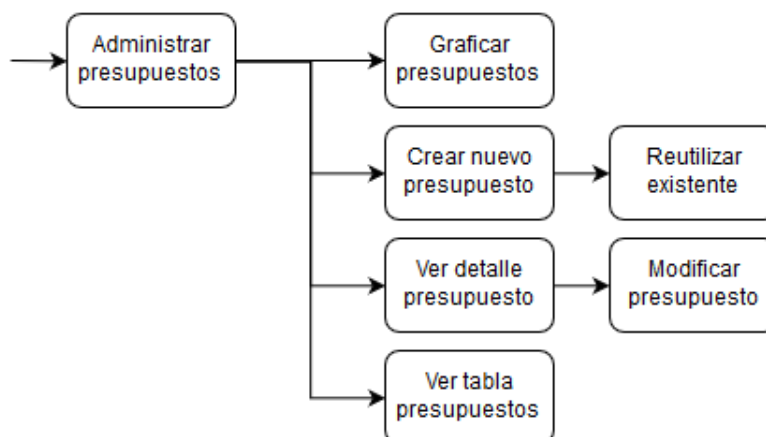


Figura 20. Módulos de administración de presupuestos.

4.3.5. Módulos de administración de listas

Encargados de la creación y modificación de listas (Figura 21).

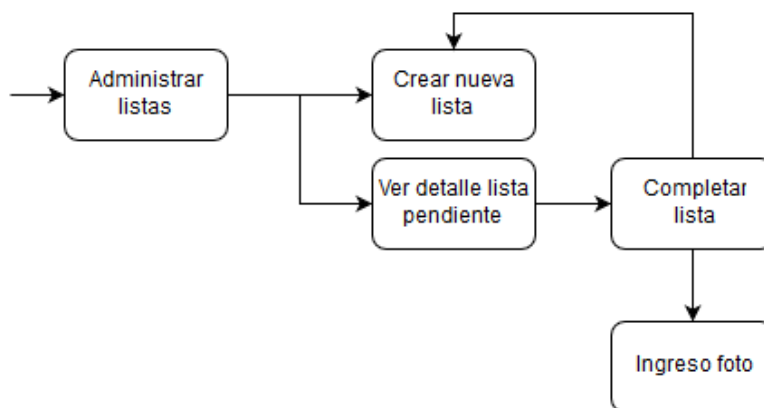


Figura 21. Módulos de administración de listas.

4.3.6. Módulos de administración de ítems

Permiten crear y modificar ítems (Figura 22).

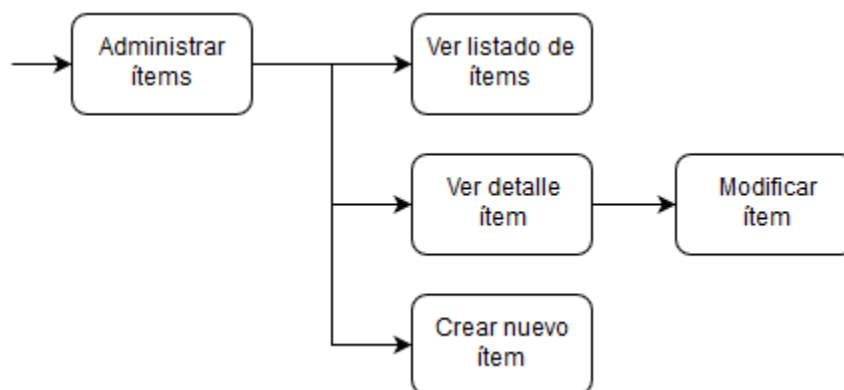


Figura 22. Módulos de administración de ítems.

4.3.7. Módulos de administración de categorías

Permiten crear, modificar y eliminar categorías asociadas a la vivienda activa (Figura 23).

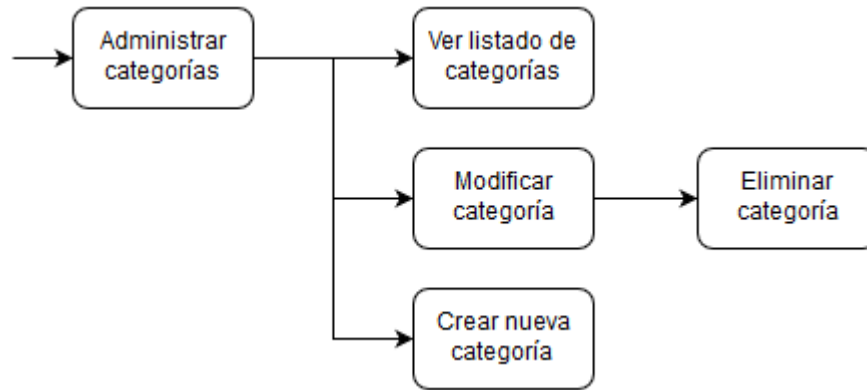


Figura 23. Módulos de administración de categorías.

4.3.8. Módulos de administración de vacaciones

Permiten crear y modificar instancias de vacaciones asociadas al usuario activo (Figura 24).

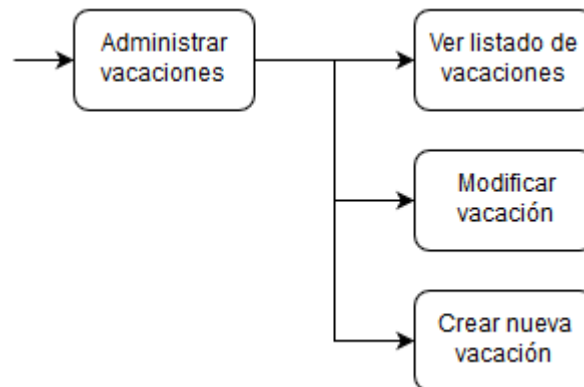


Figura 24. Módulos de administración de vacaciones.

4.4. Modelo de datos

La Figura 26 muestra las entidades correspondientes a tablas que provee Django para la administración de usuarios. Por otro lado, la Figura 25 muestra el modelo de datos relacional del sistema mismo. Notar que todas estas tablas comienzan con “*expenses_manager_*”; esto se debe a que las tablas se generan automáticamente a partir de *models.py*, y el nombre del proyecto es “*expenses_manager*”. Las tablas representadas en la Figura 26 se relacionan con las tablas presentadas en la Figura 25 a través de la entidad **auth_user**.

El detalle de las entidades se presenta a continuación:

- *auth_user*: guarda la información de los usuarios.
- *expenses_manager_vivienda*: guarda la información de las viviendas.

- *expenses_manager_viviendausuario*: relaciona usuarios con viviendas, es decir, agrupa a los usuarios que viven en una misma vivienda.
- *expenses_manager_yearmonth*: indica un mes de un año determinado.
- *expenses_manager_invitación*: identifica a una instancia de ViviendaUsuario que está invitando a otro usuario a formar parte de la vivienda asociada a ésta.
- *expenses_manager_categoría*: indica un tipo de gasto (supermercado, luz, arriendo, etc.).
- *expenses_manager_item*: indica un producto o ítem en el cual se suelen invertir fondos (manzanas, ampollitas, etc.).
- *expenses_manager_userisout*: indica un rango de tiempo en que el usuario se encuentra ausente.
- *expenses_manager_presupuesto*: asocia una categoría con una vivienda, e indica un monto presupuestado para invertir en dicha categoría en un determinado período año-mes.
- *expenses_manager_listacompras*: lista de compras creada por un usuario en una fecha determinada. El estado puede ser “completada”, “eliminada” o “pendiente”.
- *expenses_manager_itemlista*: asocia ítems con una lista de compras, indicando cuánto es necesario comprar de cada ítem. Al momento de revisar la lista de compras, se va actualizando el campo “cantidad_comprada” de cada ítem, lo que a su vez actualiza el estado de los ítems en la lista de “pendiente” a “comprado”.
- *expenses_manager_gasto*: un gasto corresponde a una inversión al fondo común que ya se hizo (estado “pagado”), o que debe hacerse en el futuro (estado “pendiente”). La entrada *añoMes* se genera automáticamente a partir de la fecha de pago. Se puede generar un Gasto con estado “pagado” al marcar como “completada” una lista de compras, en cuyo caso la categoría es “supermercado”, y el usuario activo queda registrado como quien pagó. Por otro lado, cualquier usuario puede crear un Gasto con estado “pendiente”, por ejemplo, cuando llega la cuenta de la luz. Luego, otro usuario puede marcarla como “pagado”, en cuyo caso se le asigna la inversión a dicho usuario. Finalmente, es posible saltarse este último paso y generar directamente un Gasto con estado “pagado” para cualquier categoría.
- *expenses_manager_estadogasto*: indica el estado del gasto asociado. Los estados posibles son: *pendiente*, *por confirmar* o *pagado*.
- *expenses_manager_confirmaciongasto*: cuando un gasto pasa del estado *pendiente* a *por confirmar*, se genera una instancia de *confirmacionGasto* por cada usuario activo al

momento del pago. Cada usuario debe confirmar el gasto de forma independiente, lo cual se hace cambiando el flag *confirmado* de la instancia de *confirmacionGasto* asociada al Gasto. Una vez que todas las instancias de *confirmacionGasto* relacionadas con el gasto se encuentran confirmadas, se cambia el estado del Gasto a *pagado*.

No se entrará en detalle sobre las entidades presentes en la Figura 26, ya que estas son generadas automáticamente por el framework, y tienen que ver principalmente con autenticar a los usuarios y gestionar la página de administración de Django.

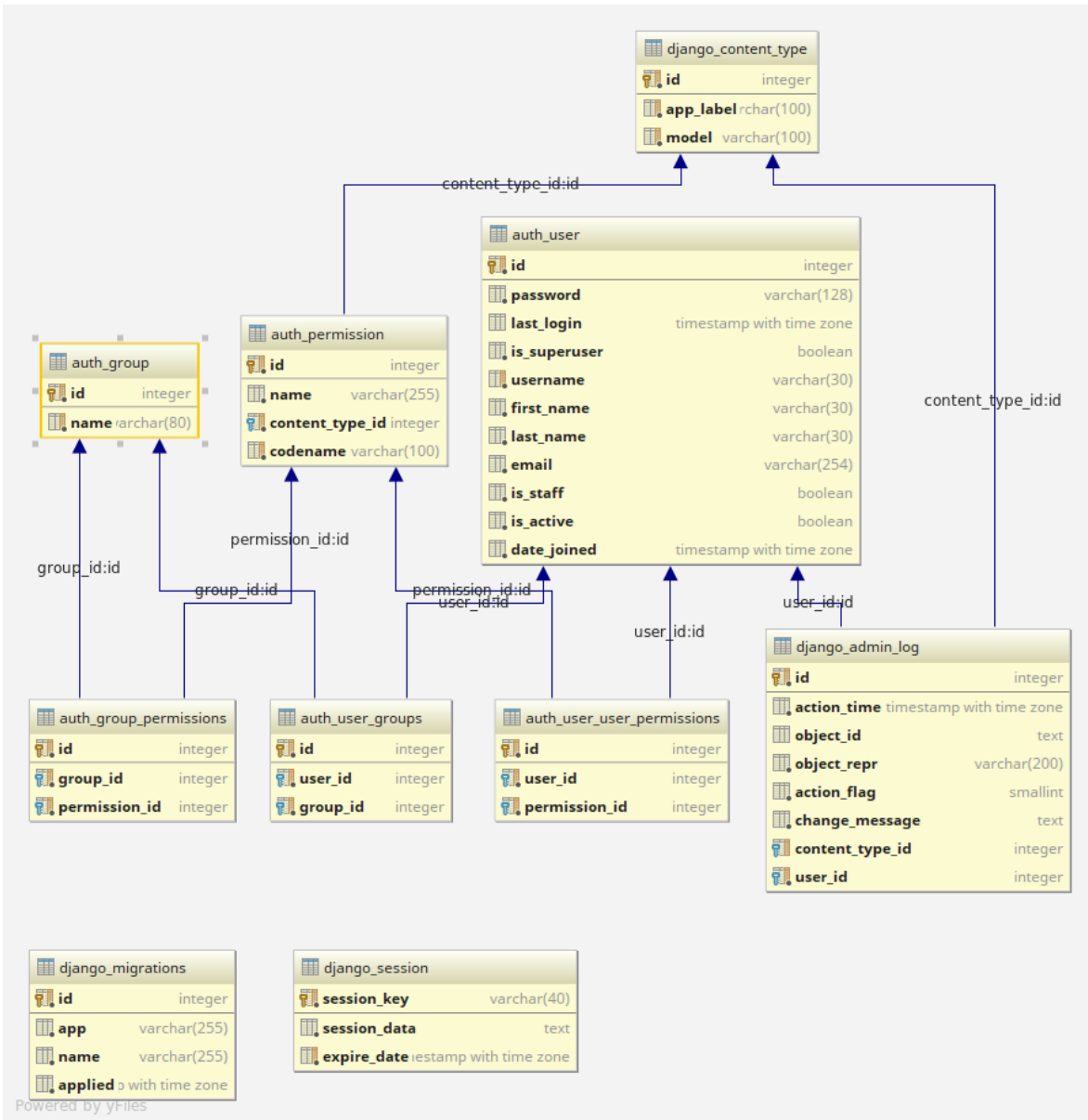


Figura 26. Entidades de Django para la administración de usuarios. Esta figura extiende el modelo presentado en la Figura 25 a través del modelo `auth_user`.

5. Implementación de la solución

A continuación, se describe la implementación de las distintas partes de la solución. Es importante notar que, si bien no existen distintos roles entre los usuarios, en algunas secciones del sitio se hace la distinción entre:

1. Usuarios no autenticados.
2. Usuarios sin vivienda activa.
3. Usuarios con vivienda activa.

Sin embargo, en la mayoría de los casos esta distinción consiste simplemente en restringirle el acceso a los datos a 1 y 2.

5.1. Sobre el template base y los CSS utilizados

Todas las interfaces de Django se generan a partir de un *template*, el cual corresponde a un archivo .HTML construido en base a variables que arrojan las vistas. Con el fin de minimizar la cantidad de código repetido, Django permite crear un template *base.html*, el cual luego es extendido por los templates específicos de cada sección de la aplicación.

De esta forma, en el template base se encuentran definidos los títulos y subtítulos, así como la barra de navegación. Los templates de Django permiten definir condiciones, por lo que la barra de navegación es capaz de adaptarse dependiendo de qué usuario se trata: un usuario que no se ha autenticado (Figura 27), un usuario sin vivienda (Figura 28) o un usuario con vivienda (Figura 29).

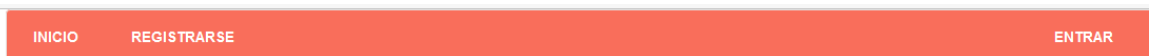


Figura 27. Barra de navegación para un usuario que no se ha autenticado.

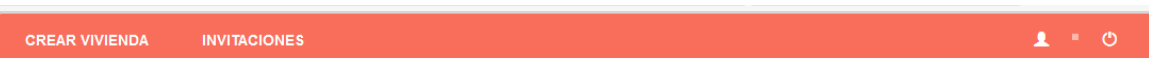


Figura 28. Barra de navegación para usuario autenticado, pero sin vivienda activa.

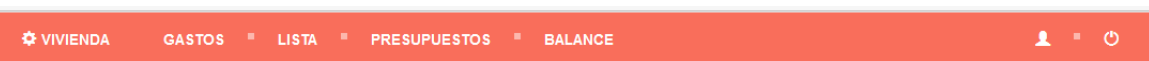


Figura 29. Barra de navegación para un usuario autenticado con vivienda activa.

Para facilitar que la aplicación funcione bien en la mayor cantidad de navegadores y dispositivos, se optó por utilizar *django-bootstrap*¹², una librería de Python que permite tener acceso a las distintas clases de *bootstrap 3*¹³. *Bootstrap* es una librería de CSS que provee muchas reglas de estilo adicionales a las propias, facilitando así que los distintos componentes de la aplicación (formularios, barras de navegación, etc) sean suficientemente responsivos.

Por otro lado, para la generación de contenido dinámico en los templates se hace uso de *django-jquery*¹⁴, que corresponde a una librería de Python que permite hacer uso de todas las funcionalidades de JQuery¹⁵, librería de Javascript que facilita la manipulación de elementos HTML, así como de las solicitudes AJAX¹⁶ al servidor. Esto es necesario tanto para poder utilizar las animaciones y validaciones de formularios de bootstrap, como para poder llevar a cabo las propias validaciones y generación de contenido dinámico, como es el caso de la generación de listas de compra, como se verá más adelante.

5.2. Registro y login

Para poder hacer uso de la aplicación, el usuario debe tener una cuenta con la cual acceder al sitio. Si bien Django provee un sistema de autenticación básico, se optó por utilizar *django-registration*¹⁷, una librería de Python que trabaja sobre las componentes de autenticación de Django, facilitando así el desarrollo del mecanismo de autenticación.

5.2.1. Registro

Django-registration permite varios métodos de registro, pero se optó por el *hmac workflow*¹⁸, el cual consiste en el registro y posterior activación por email. De esta forma, el sistema se asegura que el email ingresado por el usuario es válido, y por lo tanto podrá ser utilizado para recuperar la contraseña.

¹² <https://django-bootstrap3.readthedocs.io/en/latest/>

¹³ <http://getbootstrap.com/>

¹⁴ <https://pypi.python.org/pypi/django-jquery>

¹⁵ <https://jquery.com/>

¹⁶ http://www.w3schools.com/ajax/ajax_intro.asp

¹⁷ <http://django-registration.readthedocs.io/en/2.0.3/>

¹⁸ <http://django-registration.readthedocs.io/en/2.0.3/hmac.html>

INICIO REGISTRARSE ENTRAR

Nombre de usuario

Nombre de usuario

Requerido. 30 caracteres o menos. Letras, dígitos y @/./+/_ solamente.

Email

Email

dirección de correo electrónico

Contraseña

Contraseña

Contraseña (confirmación)

Contraseña (confirmación)

Para verificar, introduzca la misma contraseña anterior.

Registrarse!

Figura 30. Página de registro.

Notar que la validación del formulario impide ingresar un nombre de usuario o email que ya se encuentra en uso:

Nombre de usuario

jsanz

Requerido. 30 caracteres o menos. Letras, dígitos y @/./+/_ solamente. Ya existe un usuario con este nombre.

Email

jaime.sanz@ug.uchile.cl

dirección de correo electrónico This email address is already in use. Please supply a different email address.

Figura 31. Formulario impide repetir nombre de usuario y/o email.

Luego de llenar y enviar correctamente el formulario (Figura 32), el usuario recibe un correo con el link de activación (Figura 33), el cual contiene el link de activación. Al hacer click en este, se activa la cuenta y se redirige a la vista de la Figura 34.

INICIO REGISTRARSE ENTRAR

Su cuenta fue registrada, pero antes de poder usarla debe activarla. Se envió un mail con instrucciones para activarla.

Figura 32. Se envía mail de activación.

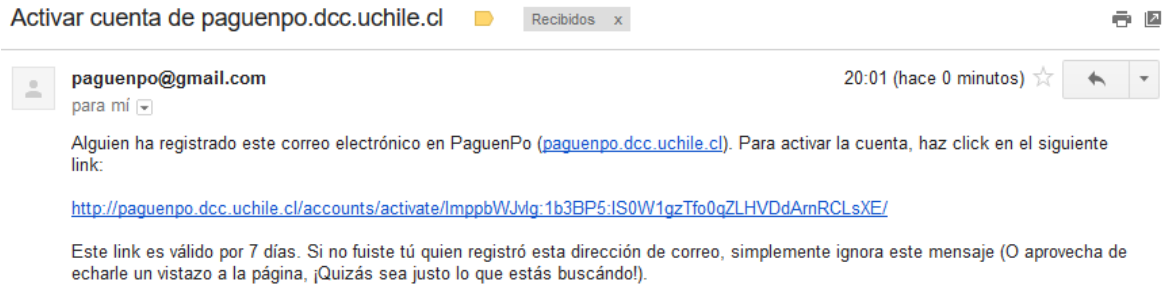


Figura 33. Mail de activación con link para activar.

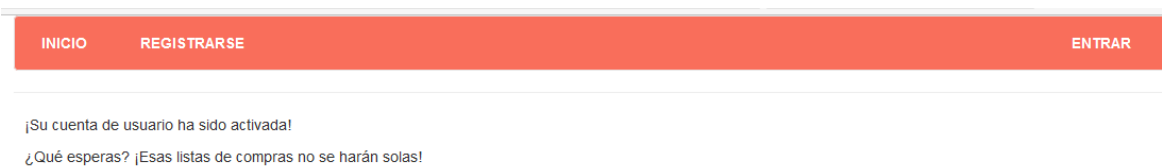


Figura 34. Cuenta activada.

5.2.2. Recuperación de contraseña

Django-registration cuenta también con un sistema de recuperación de contraseña a través del email. Este formulario es accesible desde la vista del login, como se ve en la Figura 35.



Figura 35. Link para recuperar contraseña.

El proceso de recuperación es bastante estándar: el usuario ingresa el correo asociado a su cuenta (Figura 36), para luego enviar la solicitud de recuperación (Figura 37). Luego, el usuario recibirá un correo con el link al formulario para cambiar la contraseña (Figura 38), el cual lo llevará al formulario en cuestión (Figura 39). Finalmente, el usuario cambia la contraseña y se le muestra un mensaje indicando que el proceso llegó a su fin (Figura 40).

The screenshot shows a web interface with a red header bar containing the links "INICIO", "REGISTRARSE", and "ENTRAR". Below the header, there is a section titled "Correo electrónico" with a text input field containing the placeholder "Correo electrónico". A blue button labeled "Recuperar" is positioned below the input field.

Figura 36. Formulario para recuperar contraseña.

The screenshot shows a confirmation message on a web page. At the top, there is a red header bar with "INICIO", "REGISTRARSE", and "ENTRAR". Below the header, the text reads: "Se envió un mail con instrucciones para recuperar su clave".

Figura 37. Se envía mail con link para reestablecer contraseña.

The screenshot shows an email interface. The subject line is "Contraseña restablecida en paguenpo.dcc.uchile.cl". The sender is "paguenpo@gmail.com" and the recipient is "para mí". The time is "20:12 (hace 0 minutos)". The email body contains the text: "Alguien ha solicitado recuperar la contraseña de paguenpo.dcc.uchile.cl asociada a este correo electrónico. Si fue usted, haga click en el siguiente link para reestablecer su contraseña:" followed by the URL <http://paguenpo.dcc.uchile.cl/accounts/password/reset/confirm/MTA/4c0-4dbbb5790d83cb952eb0/>. It concludes with: "Si no fue usted quien realizó esta solicitud, simplemente ignore este correo."

Figura 38. Correo con link de para reestablecer contraseña.

The screenshot shows a form for setting a new password. It has a red header bar with "INICIO", "REGISTRARSE", and "ENTRAR". The section is titled "Contraseña nueva" and has a text input field. Below the field are four bullet points: "Su contraseña no puede asemejarse tanto a su otra información personal.", "Your password must contain at least 8 characters.", "Your password can't be a commonly used password.", and "Su contraseña no puede ser completamente numérica.". Below these points is another section titled "Contraseña nueva (confirmación)" with a text input field. A blue button labeled "Reestablecer" is at the bottom.

Figura 39. Formulario para reestablecer contraseña.

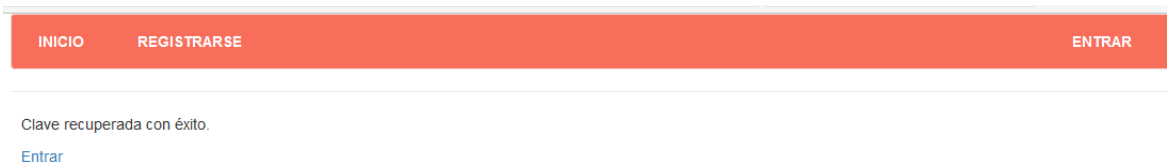


Figura 40. Contraseña reestablecida exitosamente.

5.2.3. Login

Los usuarios pueden ingresar a la plataforma haciendo click en el botón “Entrar” en la barra de navegación. Esto los llevará al formulario de ingreso (Figura 41).

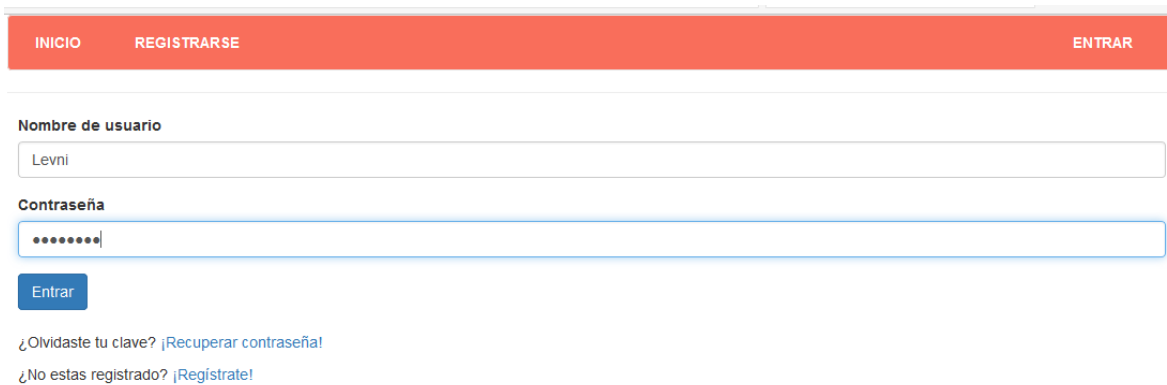


Figura 41. Formulario de ingreso a la plataforma.

Una vez que se ingresa a la plataforma, existen 2 páginas de inicio a las que se puede redirigir al usuario:

- Si el usuario no tiene una vivienda activa, se redirige a la página de la Figura 42.
- Si el usuario sí tiene vivienda activa, se redirige a la página de la Figura 43.

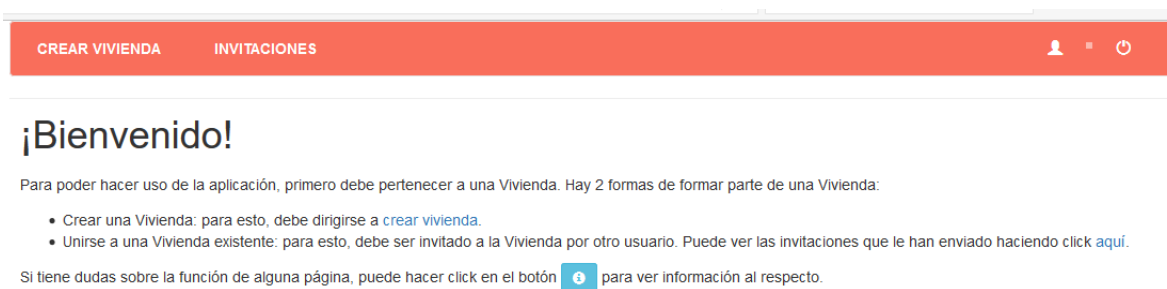


Figura 42. Página de inicio para usuario sin vivienda activa.

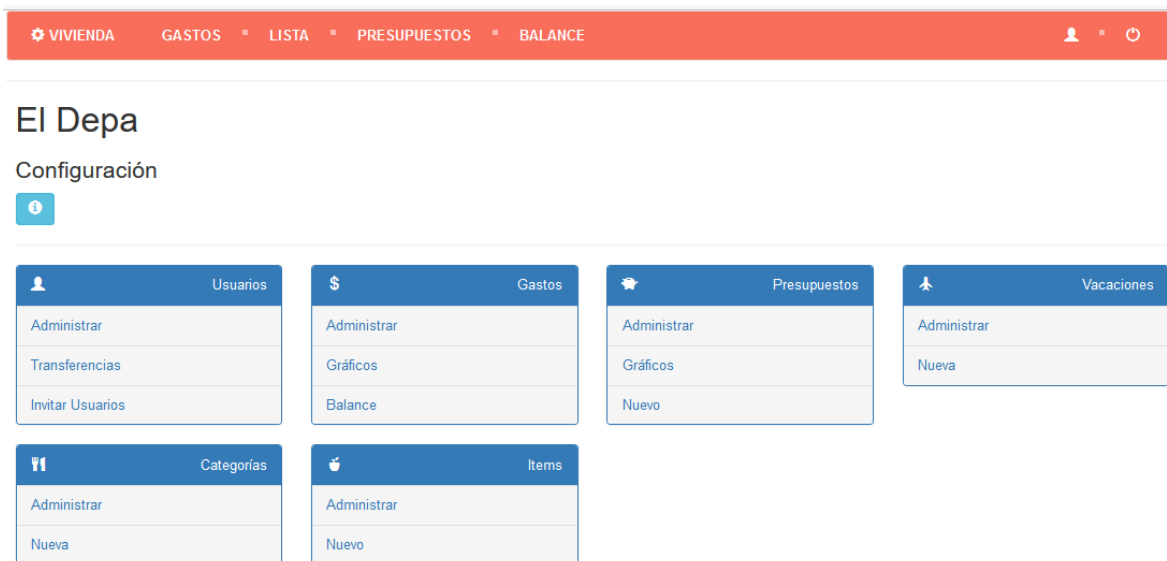


Figura 43. Página de inicio para usuario con vivienda activa. Corresponde a la página de configuración de la vivienda.

5.3. Cómo formar parte de una vivienda

Para poder hacer uso de la aplicación, el usuario debe pertenecer a una vivienda. Para esto, existen dos alternativas: crear una nueva vivienda o bien unirse a una vivienda ya existente.

5.3.1. Crear una nueva vivienda

Los usuarios sin vivienda tienen la opción de crear una nueva vivienda; esta se genera con una lista básica de ítems y categorías, de forma tal de minimizar la cantidad de configuraciones previas que deben hacerse a la vivienda antes de comenzar a utilizarla.

Para crear una nueva vivienda, el usuario hace click en el botón “Crear Vivienda” en la barra de navegación, lo que lo llevará al formulario de creación de viviendas (Figura 44). Si el formulario no contiene errores, se redirige al usuario a la vista de la Figura 45.

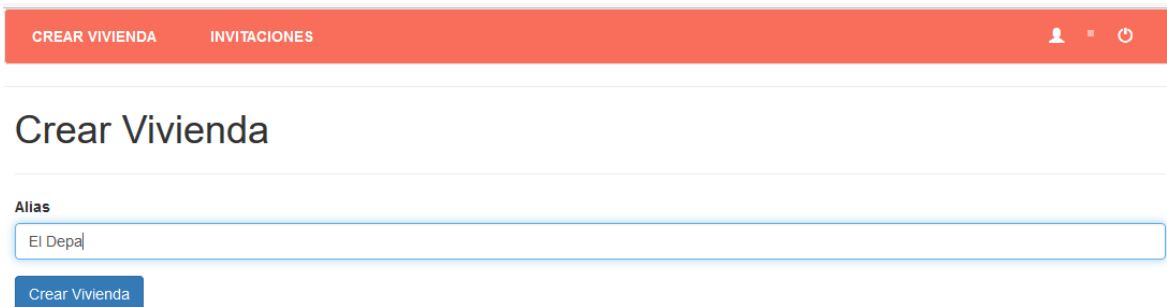


Figura 44. Formulario para crear una nueva vivienda.



Figura 45. Luego de crear la vivienda, se redirige al usuario a la página de configuración de la nueva vivienda.

5.3.2. Unirse a vivienda existente

Como se verá en la sección 5.5, un usuario que pertenece a una vivienda puede invitar a otros usuarios a formar parte de ésta. En esta sección se verán las interfaces que permiten que el usuario invitado acepte dicha invitación, y de esta forma pase a formar parte de la vivienda ya existente.

La Figura 46 muestra la lista de invitaciones recibidas por el usuario. En la Figura 47, el usuario decide si quiere aceptar o rechazar la invitación, y la Figura 48 muestra al usuario en la página de configuración de la vivienda, luego de aceptar la invitación.

Notar que el usuario inmediatamente tiene acceso a las categorías e ítems personalizados definidos para esta vivienda.

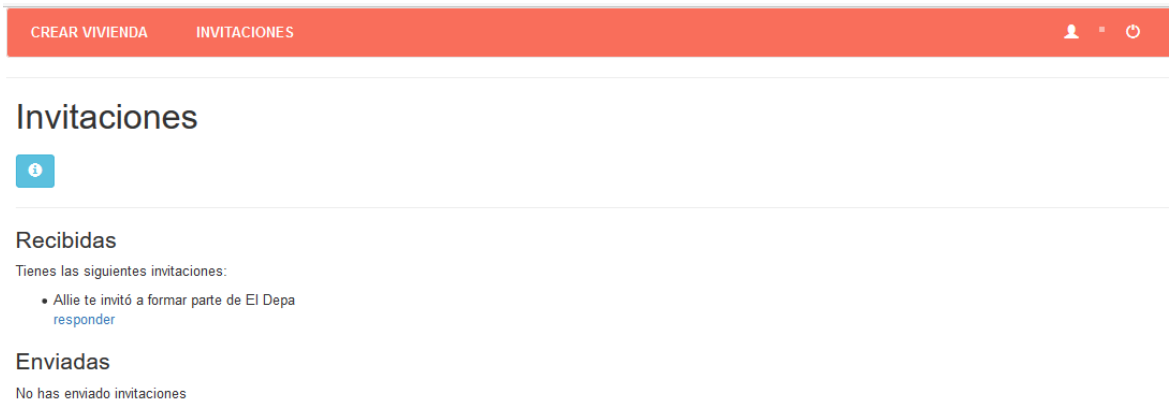


Figura 46. Lista de invitaciones, donde se ve una invitación de parte del usuario *Allie* para formar parte de la vivienda *El Depa*.

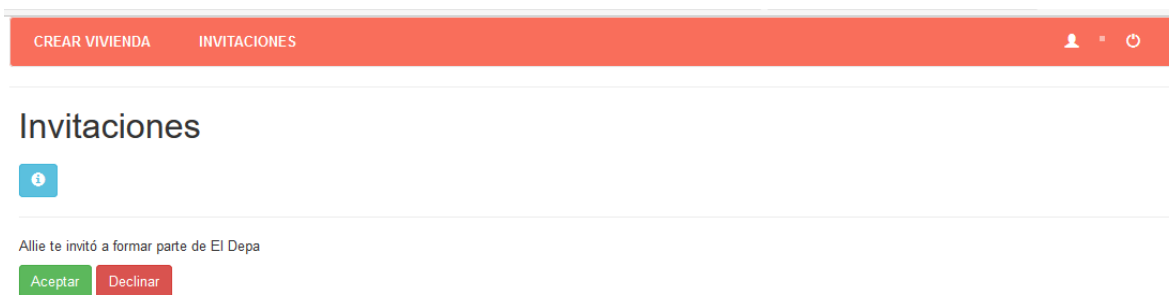


Figura 47. Detalle invitación recibida.

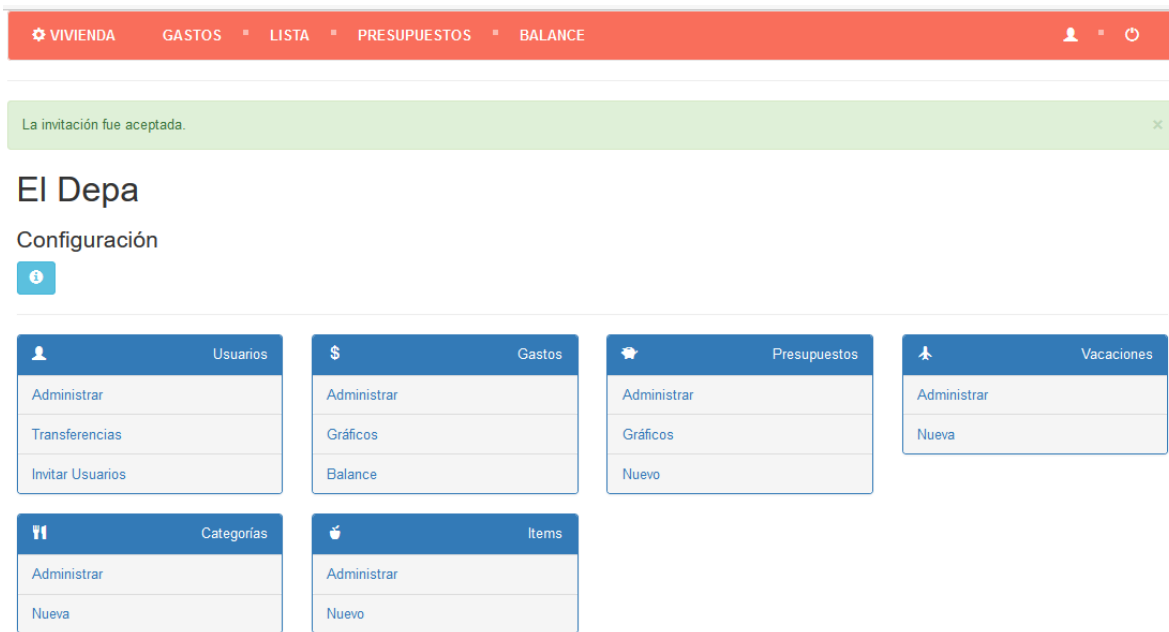


Figura 48. Al aceptar la invitación, se redirige a la configuración de la vivienda.

5.4. Sobre la página de configuración de la vivienda

La página de configuración de la vivienda (Figura 49) corresponde al punto central de la aplicación desde donde los usuarios tienen acceso de forma directa a casi todas las funcionalidades del proyecto.

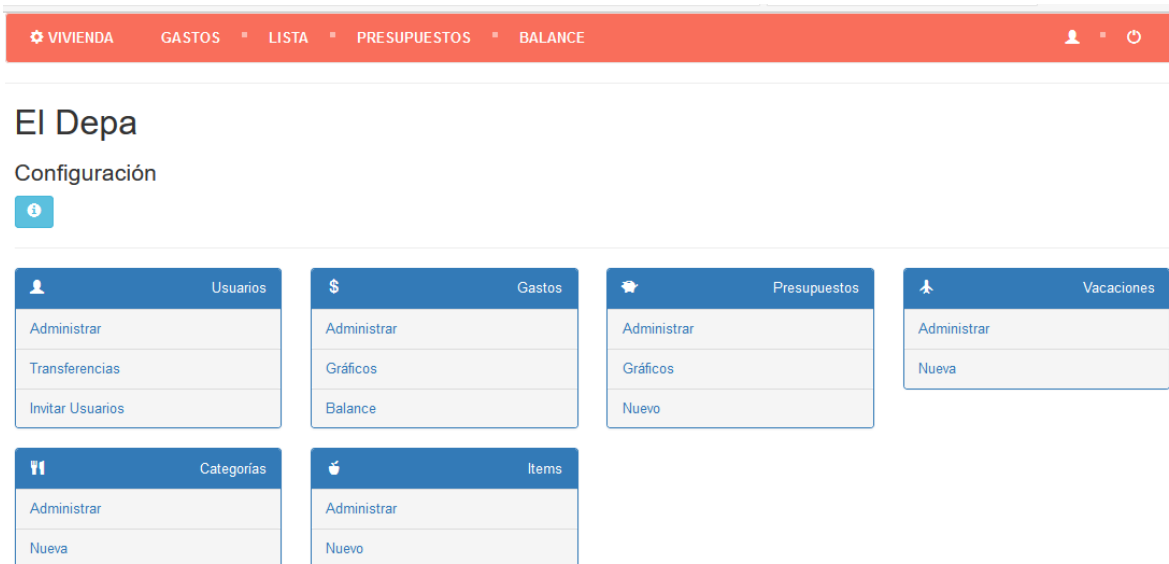


Figura 49. Página de configuración de la vivienda.

Esta interfaz, al igual que las interfaces de las partes centrales de la aplicación¹⁹, cuenta con un mensaje de ayuda (Figura 50), el cual sugiere el orden en que los nuevos usuarios debiesen leer los mensajes de ayuda, de tal forma de hacerse una idea de cómo hacer buen uso de la aplicación.

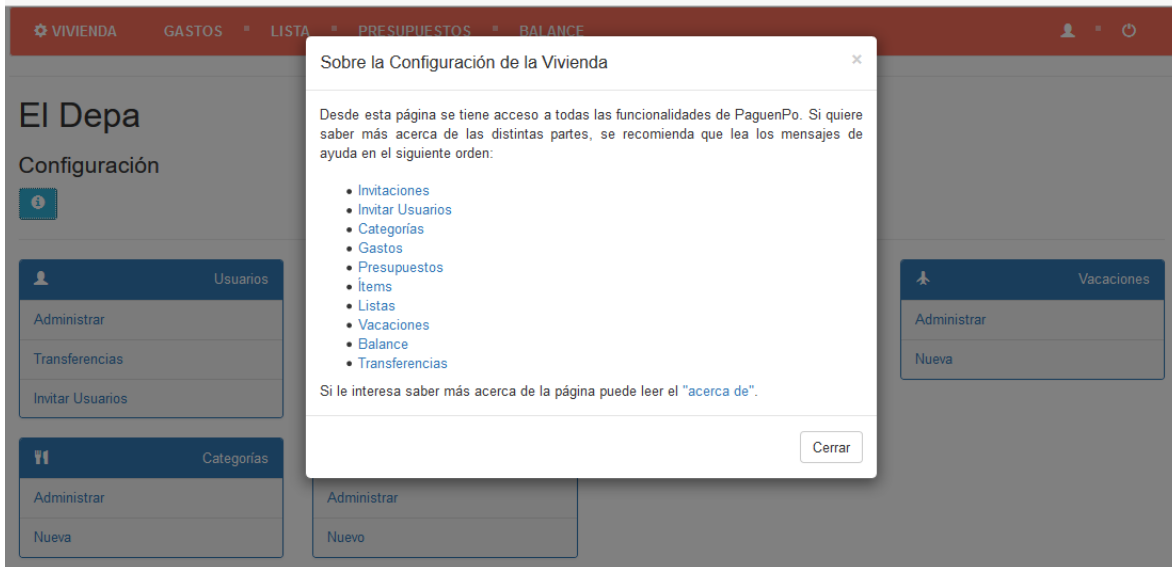


Figura 50. Mensaje de ayuda de la página de configuración.

5.5. Invitar usuarios

Cualquier usuario perteneciente a una vivienda tiene la capacidad de invitar nuevos usuarios a esta, los cuales luego pueden responder a las invitaciones, como se vio en la sección 5.3.2.

Para realizar invitaciones, los usuarios se dirigen a la página de configuración (Figura 49), desde donde se tiene acceso al link "Invitar Usuarios". Seguidamente, los usuarios deberán llenar el formulario de invitación (Figura 51), lo cual a su vez generará una nueva invitación, como puede verse en la Figura 52. Por su parte, el usuario que está siendo invitado recibe la invitación y puede responder a esta, como ya se vio anteriormente.

¹⁹ Dígase: gastos, categorías, ítems, presupuestos, listas, vacaciones y balance

VIVIENDA GASTOS LISTA PRESUPUESTOS BALANCE

El Depa

Invitar Usuario

Info

Email

Invitar

Figura 51. Formulario para invitar usuarios a la vivienda.

VIVIENDA GASTOS LISTA PRESUPUESTOS BALANCE

Invitaciones

Info

Recibidas

No has recibido invitaciones

Enviadas

Has realizados las siguientes invitaciones:

- Has invitado a Levni a formar parte de El Depa [ver](#)

Figura 52. Lista de invitaciones enviadas.

El usuario que genera la invitación tiene la capacidad de cancelar sus invitaciones enviadas (Figura 53), mientras que los usuarios que reciben las invitaciones pueden aceptar o declinar, como ya se vio anteriormente (Figura 47).

VIVIENDA GASTOS LISTA PRESUPUESTOS BALANCE

Invitaciones

Info

Has invitado a Levni a formar parte de El Depa

Cancelar

Figura 53. Usuario puede cancelar invitación enviada.

5.6. Manejo de categorías

Las categorías corresponden al pilar fundamental de la aplicación, pues todas las funcionalidades tienen relación con los gastos, los cuales a su vez están contruidos sobre las categorías. En las

secciones siguientes se describen todas las interacciones que los usuarios pueden tener con éstas, y cómo estas interacciones afectan la forma en que el resto de la aplicación percibe los gastos, presupuestos y balances asociados a la vivienda. Al ser una parte tan fundamental del proyecto, todas las interacciones con las categorías parten desde la página de configuración de la vivienda (Figura 49).

5.6.1. Categorías por *default*

Al crear una nueva vivienda, esta viene con un cierto conjunto de categorías por *default*, las cuales pueden ser editadas²⁰ por cualquier usuario de la vivienda. Estos cambios a las categorías son percibidos inmediatamente por todos los usuarios de la vivienda, y sólo por los usuarios de la vivienda, pues las categorías son propias de cada vivienda independiente.

La lista de categorías se accede al hacer click en “Administrar”, en la sección de “Categorías” de la página de configuración. La lista de categorías puede verse en la Figura 54. En este listado pueden verse también los atributos de las distintas categorías.

El Depa

Categorías

+ Nueva
⚠ Ver botón eliminar
ⓘ

Categoría ▲	Ocultas ⇅	Compartida ⇅	Compartida en Vacación ⇅
Agua	✘	✓	✓
Arriendo	✘	✓	✓
Cable	✘	✓	✓
Gas	✘	✓	✓
Gastos Comunes	✘	✓	✓
Internet	✘	✓	✓
Luz	✘	✓	✓
Otros	✘	✓	✓
Servicios	✘	✓	✓
Supermercado	✘	✓	✓
Teléfono	✘	✓	✓

Figura 54. Listado de categorías por *default* de la vivienda.

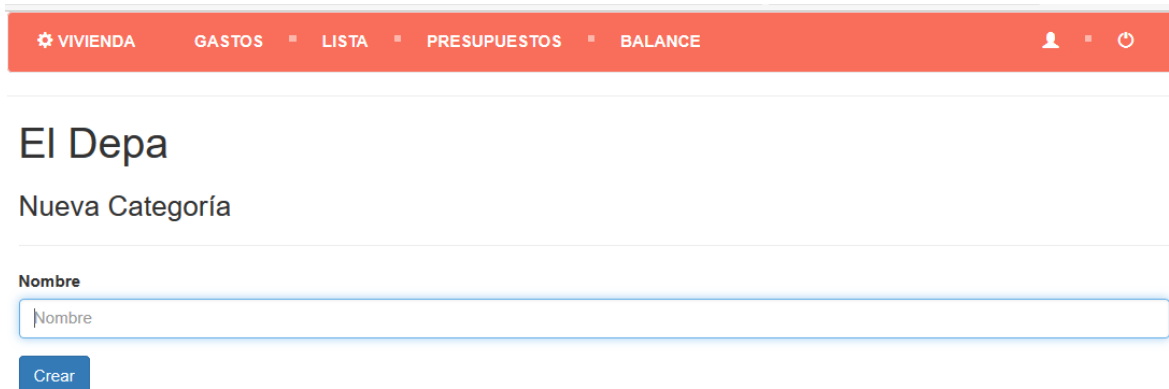
Por defecto, las categorías **no** se encuentran *ocultas*, y son todas tanto *compartidas* como *compartidas en vacación*. Notar que, al igual que muchas otras secciones, se hace uso de un plugin

²⁰ Estas categorías pueden ser modificadas, pero no pueden ser eliminadas.

de JQuery llamado *DataTables*²¹ que facilita la construcción de tablas responsivas y con paginación.

5.6.2. Crear categoría

Los usuarios pueden crear categorías personalizadas dependiendo de las necesidades propias de la vivienda. Para esto, los usuarios pueden hacer click en “Nueva” en la sección de “Categorías” de la página de administración, o bien hacer click en el botón “Nueva” de la página de categorías (Figura 54), lo cual despliega la interfaz que se muestra a continuación.



The screenshot shows a web interface for 'El Depa'. At the top, there is a red navigation bar with the following menu items: VIVIENDA, GASTOS, LISTA, PRESUPUESTOS, and BALANCE. On the right side of the bar are icons for a user profile and a refresh button. Below the navigation bar, the page title is 'El Depa' and the sub-header is 'Nueva Categoría'. The form consists of a text input field labeled 'Nombre' with a placeholder 'Nombre' and a blue 'Crear' button below it.

Figura 55. Formulario para crear nueva categoría.

La única condición para la creación de nuevas categorías es que no se repitan los nombres, como puede verse en la Figura 56.

²¹ <https://datatables.net/>

El nombre ingresado corresponde a una categoría global 

El Depa

Nueva Categoría

Nombre

Crear

Figura 56. Mensaje de alerta al intentar crear categoría personalizada con nombre “Agua”. La validación se hace en el lado del servidor, por lo es necesario enviar el formulario para obtener el mensaje.

Si el usuario ingresa un nombre de categoría personalizada (Fig. 57) y éste cumple la condición mencionada anteriormente, entonces la categoría es creada correctamente, y se hace visible para todos los usuarios de la vivienda, como puede verse en la Figura 58.

El Depa

Nueva Categoría

Nombre

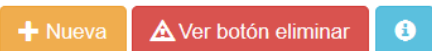
Crear

Figura 57. Usuario ingresa nombre de categoría personalizada.

¡Categoría agregada!

El Depa

Categorías



Categoría	Oculto	Compartida	Compartida en Vacación
Teléfono	✘	✔	✔
Supermercado	✘	✔	✔
Servicios	✘	✔	✔
Personal_1	✘	✔	✔
Otros	✘	✔	✔

Figura 58. El usuario es redirigido al listado de categorías luego de ingresar categoría personalizada. La categoría ingresada puede verse en el listado.

5.6.3. Categorías especiales

Existen distintos tipos de categoría, las cuales cumplen propósitos distintos y modifican la forma en que se calculan tanto los balances como los totales por período. Existen 3 categorías especiales las cuales están asociadas a funciones específicas de la aplicación, y por lo tanto están necesariamente presentes en todas las viviendas. Estas categorías corresponden a:

- *Supermercado*: al completar una lista de compras, el gasto generado automáticamente tiene asociada esta categoría.
- *Transferencias*: al hacer una transferencia a otro usuario, los gastos generados están asociados a esta categoría. Esta categoría no puede ser editada por los usuarios, razón por la cual no es posible verla en la Figura 54.
- *Otros*: para efectos de cálculos de totales, se consideran de esta categoría los gastos que:
 - o Están directamente asociados a la categoría "Otros".
 - o Su categoría asociada se encuentra *oculta* (ver sección 5.6.4).

5.6.4. Tipos de categorías

Los usuarios pueden modificar los atributos de las categorías, por lo que surgen 3 tipos de categorías:

- *Ocultas*: para efectos de cálculos de totales y presupuestos, los gastos asociados a una categoría oculta son considerados como si fuesen parte de la categoría “Otros”.
- *Compartidas*: los gastos asociados a este tipo de categoría son los que se consideran para el cálculo de²²:
 - Totales por período por categoría
 - Totales de los presupuestos
 - Balance de los usuarios
- *Compartidas en vacación*: para efectos del cálculo de balances de los usuarios, los gastos asociados a este tipo de categoría siempre se consideran para el total esperado por usuario.

5.6.5. Eliminar categoría

Los usuarios pueden eliminar cualquier categoría personalizada. Los gastos asociados a dicha categoría pasan a estar relacionados con la categoría “Otros”. Para eliminar una categoría, los usuarios deben hacer click en el botón “Ver botón eliminar” presente en la interfaz del listado de categorías. Como puede verse en la Figura 59, luego de hacer click se revela una columna adicional en la tabla. Notar que las categorías *default* no pueden ser eliminadas, por lo que, en lugar del ícono para eliminar, aparece una **X**. Notar además que ahora se da la opción de “Esconder botón Eliminar”.

²² Actualmente no se está haciendo uso de las categorías que no se encuentran compartidas. Este atributo está pensado para extender la aplicación a otros casos de uso, como se verá en la sección 7.

Categorías

+ Nueva Esconder botón Eliminar i


Categoría	Oculto	Compartida	Compartida en Vacación	Eliminar
Teléfono	✘	✔	✔	✘
Supermercado	✘	✔	✔	✘
Servicios	✘	✔	✔	✘
Personal_1	✘	✔	✔	
Otros	✘	✔	✔	✘

Figura 59. Se muestra ícono para eliminar categoría. No es posible eliminar categorías default.

Seguidamente, deben hacer click en el ícono de eliminar de la categoría que se desea eliminar. El usuario es redirigido a la misma interfaz, como puede verse en la Figura 60.

Categoría eliminada. Los gastos asociados a ésta se traspasaron a la categoría 'Otros'

El Depa

Categorías

+ Nueva Ver botón eliminar i

Categoría	Oculto	Compartida	Compartida en Vacación
Teléfono	✘	✔	✔
Supermercado	✘	✔	✔
Servicios	✘	✔	✔
Otros	✘	✔	✔

Figura 60. La categoría fue eliminada. Notar que la categoría “Personal_1” no aparece en la tabla, pues esta debiese encontrarse entre “Servicios” y “Otros”, como se vio en la Figura 59.

5.6.6. Modificar Categoría

Los usuarios pueden editar las categorías haciendo click en el nombre de éstas, en la lista de categorías de la vivienda. El formulario de edición es el siguiente (Figura 61 y Figura 62):

El Depa

Editar Categoría: Agua

- Compartida
- Compartida en Vacación
- Oculta

Guardar cambios

Figura 61. Formulario de edición de categoría.

- Compartida
- Compartida en Vacación
- Oculta

Guardar cambios

Figura 62. Se editan los atributos de la categoría.

Luego de editar exitosamente una categoría, se redirige al listado de categorías, donde pueden verse los cambios (Figura 63):

Categorías

- + Nueva
- ⚠ Ver botón eliminar
- ℹ

Categoría	Oculta	Compartida	Compartida en Vacación
Agua	✓	✗	✗
Arriendo	✗	✓	✓
Cable	✗	✓	✓

Figura 63. Se editan los campos de la categoría “Agua”, y los cambios se ven reflejados en el listado de categorías.

Los usuarios pueden modificar los atributos de las categorías, siguiendo sólo una condición: para poder marcar la categoría como *compartida en vacación*. Ésta también debe estar marcada como

compartida (Figura 64). Si se trata de infringir esta condición, se muestra el mensaje de error apropiado, como puede verse en la Figura 65.

Editar Categoría: Arriendo

- Compartida
- Compartida en Vacación
- Oculta

Guardar cambios

Figura 64. Combinación inválida de atributos de la categoría.

Si la categoría no es compartida, no puede ser marcada como compartida en vacación. ×

- Compartida
- Compartida en Vacación
- Oculta

Guardar cambios

Figura 65. Mensaje de error luego de introducir combinación inválida de atributos de la categoría.

5.7. Manejo de gastos

5.7.1. Sobre el proceso de confirmación de gastos

Con el fin de transparentar los gastos ingresados por los usuarios de la vivienda, se define un proceso de *confirmación* de los gastos. Es decir, antes de que los gastos ingresados por los usuarios sean considerados para efectos de cálculos de totales invertidos en la vivienda, estos deben ser confirmados por todos los usuarios de la vivienda. La Figura 66 muestra este proceso.

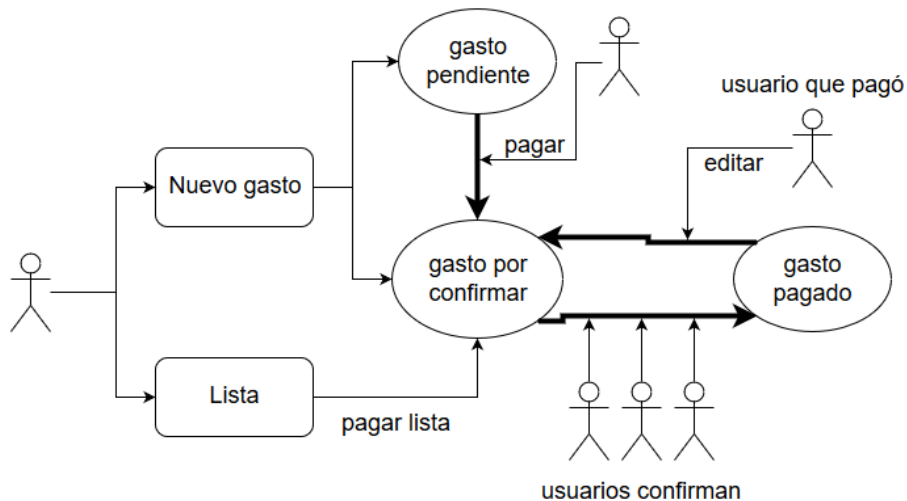


Figura 66. Proceso de confirmación de gastos. Las flechas en negrita representan un cambio de estado del gasto, el cual es gatillado por la acción indicada: pagar, confirmar o editar. Notar que el usuario que inicia el proceso y el que paga el gasto pendiente no son necesariamente el mismo.

Existen 2 formas de crear un nuevo gasto: (1) puede hacerse de forma directa, o (2) se realiza automáticamente al momento de pagar una lista de compras. Si el gasto se crea con estado pendiente, este puede ser pagado por cualquier usuario de la vivienda, no necesariamente por el usuario que lo creó.

El gasto es considerado como *pagado* una vez que es confirmado por todos los usuarios de la vivienda. Notar que, al momento de pagar un gasto, el usuario que paga automáticamente confirma el pago del gasto, pues no tiene sentido exigirle a este usuario que confirme manualmente. De esta forma, en una vivienda con sólo un usuario, los gastos pasan directamente al estado *pagado*.

Por último, sólo el usuario que pagó el gasto tiene permitido hacer modificaciones. Sin embargo, estas modificaciones deben ser nuevamente confirmadas por los usuarios de la vivienda antes de marcar el gasto modificado como *pagado*.

5.7.2. Crear gasto de forma directa

Como se vio en la sección anterior, los usuarios pueden crear directamente gastos pendientes o pagados. Para crear un gasto pendiente, pueden hacer click en *Gastos* en la barra de navegación, o en la página de configuración de la vivienda en *Gastos -> Administrar* (Figura 67).

El Depa

Gastos



Nuevo Gasto Pendientes Por Confirmar Pagados

Categoría
Internet

Monto
1234

Foto
Foto

Pendiente Pagado Generar

Figura 67. Formulario para crear nuevo gasto. Notar que se tiene seleccionada la opción *Pendiente*.

Se da la opción de ingresar una foto del gasto pendiente (por ejemplo, puede ser la foto de la cuenta), pero está permitido dejar este campo en blanco, no así los campos de *Categoría* y *Monto*. Al hacer click en *Generar*, se pide al usuario que revise los datos antes de confirmar la creación del gasto (Figura 68).

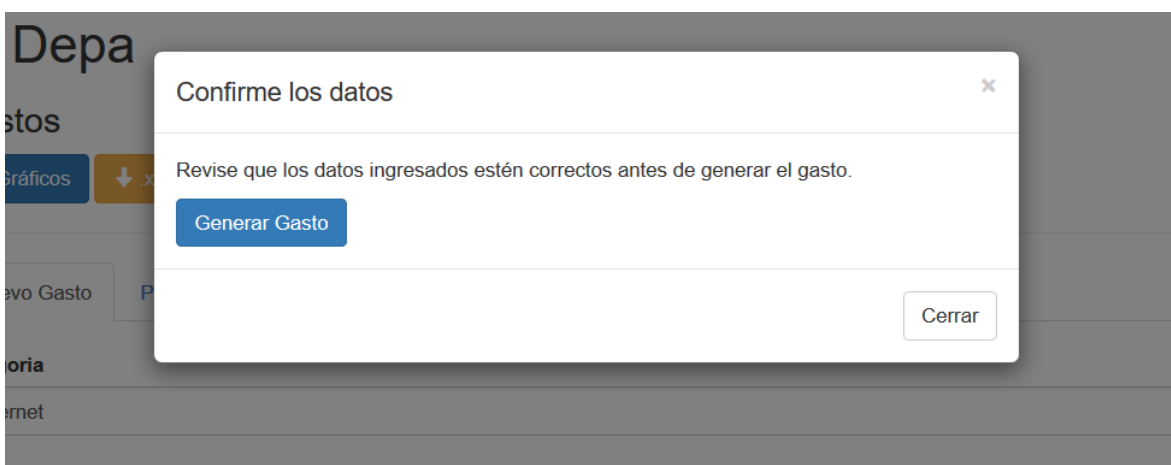


Figura 68. Se pide confirmación del usuario antes de crear el gasto.

Al confirmar los datos, se redirige a la misma página, donde puede verse un mensaje diciendo que se creó el gasto. Si se hace click en la pestaña *Pendientes*, puede verse el gasto recién creado (Figura 69).

El gasto fue creado exitosamente

El Depa

Gastos

Gráficos .xls

Nuevo Gasto Pendientes Por Confirmar Pagados

Mostrar 10 registros Buscar:

Fecha	Categoría	Monto
8 de Junio de 2016	Internet	\$ 1.234

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

Figura 69. Nuevo gasto pendiente.

Al hacer click en el ícono junto a la fecha de creación del gasto, puede verse el detalle del mismo (Figura 70).

El Depa

Detalle Gasto

Editar

Monto:	\$ 1.234
Categoría:	Internet

Pagar

Figura 70. Detalle de gasto pendiente.

Al hacer click en *Pagar*, el gasto cambia su estado a *por confirmar*, y se redirige a la misma página del detalle, donde puede verse que este ha cambiado.

Detalle Gasto



Monto:	\$ 1.234
Creado por:	Levni
Usuario:	Levni
Categoría:	Internet
Fecha creacion:	8 de Junio de 2016
Fecha pago:	8 de Junio de 2016
Periodo:	2016-6
Estado:	pendiente_confirmacion
Confirmado por:	Levni
Por confirmar:	Allie

Figura 71. Detalle de gasto con estado *por confirmar*. El campo *Usuario* indica al usuario que pagó el gasto.

En la Figura 71 se puede ver que el gasto fue pagado por el usuario *Levni*, y hace falta que lo confirme el otro usuario de la vivienda, *Allie*, antes de pasar al estado *pagado*. Sin embargo, si lo que el usuario *Levni* tiene en mente es pagar inmediatamente el gasto, tiene la opción de marcarlo como *pagado* en el formulario de nuevo gasto (Figura 72).

Un formulario web con pestañas: 'Nuevo Gasto', 'Pendientes', 'Por Confirmar' y 'Pagados'. El 'Pagados' está seleccionado. Campos: 'Categoría' (Internet), 'Monto' (1234), 'Foto' (botón naranja), 'Fecha pago' (2016-06-08). Botones: 'Pendiente', 'Pagado' (seleccionado) y 'Generar'.

Figura 72. Formulario para nuevo gasto pagado.

Al marcar la opción *pagado*, se despliega un campo adicional en el formulario: la fecha de pago del gasto. Se tiene la única condición de que la fecha indicada debe ser igual o anterior a la fecha actual.

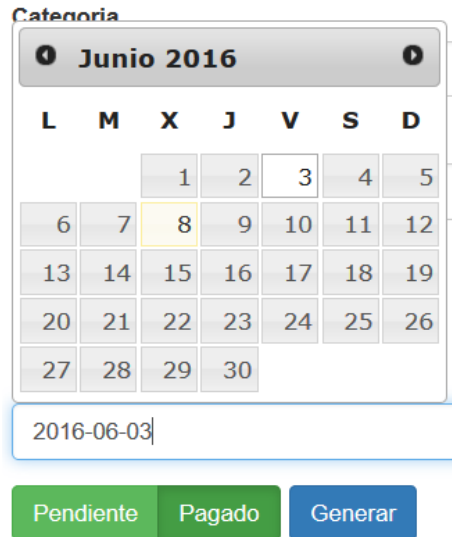


Figura 73. *Datepicker* para facilitar el ingreso de fechas. La fecha en verde claro (3) indica la fecha seleccionada, mientras que la amarilla (8) indica el día actual.

Para facilitar el correcto ingreso de fechas en la aplicación, se utiliza la librería de Django *django-jquery-ui*²³, la cual permite hacer uso de las funciones y CSS de *jquery-ui*²⁴, incluido el *datepicker* que se ve en la figura anterior. Para este ejemplo, el usuario *Levni* ingresará, además, una foto de la boleta (Figura 74).

²³ <https://pypi.python.org/pypi/django-jquery-ui/1.11.4.1>

²⁴ <https://jqueryui.com/>

[Nuevo Gasto](#)
[Pendientes](#)
[Por Confirmar](#)
[Pagados](#)

Categoría

Teléfono

Monto

123

Foto

photo_2016-06-08_18-25-20.jpg

Fecha pago

2016-06-03

[Pendiente](#)
[Pagado](#)
[Generar](#)

Figura 74. Datos para generar nuevo gasto pagado.

Luego de generar el gasto, se redirige a la misma página, y puede verse el nuevo gasto en la pestaña *por confirmar* (Figura 75).

El gasto fue creado exitosamente

El Depa

Gastos

[Gráficos](#)
[.xls](#)
[i](#)

[Nuevo Gasto](#)
[Pendientes](#)
[Por Confirmar](#)
[Pagados](#)

Mostrar registros

Buscar:

Fecha	Categoría	Monto
8 de Junio de 2016	Teléfono	\$ 123
8 de Junio de 2016	Internet	\$ 1.234

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior Siguiente

Figura 75. Listado de gastos por confirmar. Puede verse también el gasto de la Figura 69.

El detalle de este gasto (Figura 76) es igual al de la Figura 71, con la excepción de la inclusión de la foto.

Monto:	\$ 123
Creado por:	Levni
Usuario:	Levni
Categoría:	Teléfono
Foto:	ver foto
Fecha creacion:	8 de Junio de 2016
Fecha pago:	3 de Junio de 2016
Periodo:	2016-6
Estado:	pendiente_confirmacion
Confirmado por:	Levni
Por confirmar:	Allie

Figura 76. Detalle de gasto con estado *por confirmar* con foto de boleta.

Al hacer click en *ver foto* se abre una nueva pestaña donde puede verse la imagen ingresada. Esta puede tener un tamaño máximo de 500x500 pixeles. Si se ingresa una imagen más grande, se hace un *resize* para ajustarse a estas limitaciones. Experimentalmente se vio que este tamaño es suficientemente grande como para distinguir texto escrito, como puede verse en la siguiente figura.

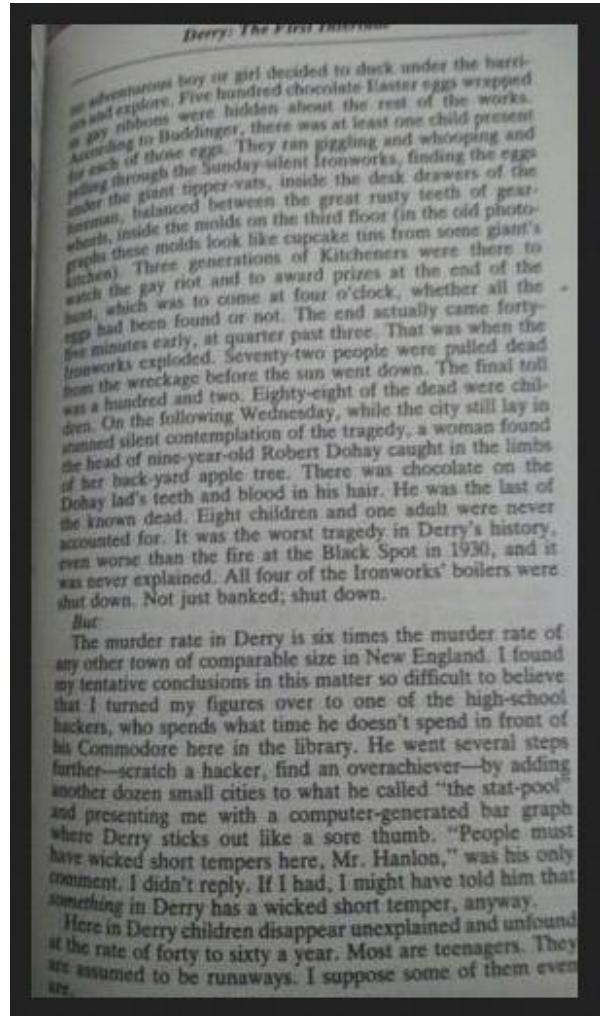


Figura 77. Izquierda: Foto de la boleta del gasto de la Figura 76 luego del resize. La foto fue tomada con un smartphone *moto G 2015*, y tenía una resolución de 720x1280 pixeles antes del resize hecho por el servidor.

Derecha: Para enfatizar que el tamaño de la imagen es suficiente, se muestra una foto de una página cualquiera de la edición de bolsillo del libro *It* de Stephen King. La foto fue tomada con el mismo smarthpone. La fuente del libro resulta considerablemente más pequeña que la de la mayoría de las boletas emitidas.

5.7.3. Crear gasto a partir de lista de compras

Al momento de pagar una lista de compras, se genera automáticamente un gasto por el monto señalado en el formulario de la lista, el cual tiene estado *por confirmar* y categoría *Supermercado*. Este gasto se diferencia de los gastos creados de forma directa en que tiene asociada la lista con los ítems comprados. El proceso completo de pagar una lista de compras se verá más adelante.

5.7.4. Confirmar gasto

Para que un gasto sea considerado para los cálculos de totales invertidos en la vivienda, debe ser previamente confirmado por todos los usuarios de la vivienda. Para esta sección, se supondrá que se tienen los gastos de la Figura 75, y que el usuario activo es *Allie*. Recordemos que ambos gastos fueron pagados por el otro usuario de la vivienda, *Levni*. Si *Allie* ingresa al detalle de los gastos, verá una interfaz muy similar a la que veía *Levni* (Figura 78).



Monto:	\$ 123
Creado por:	Levni
Usuario:	Levni
Categoría:	Teléfono
Foto:	ver foto
Fecha creacion:	8 de Junio de 2016
Fecha pago:	3 de Junio de 2016
Periodo:	2016-6
Estado:	pendiente_confirmacion
Confirmado por:	Levni
Por confirmar:	Allie



Figura 78. Detalle de gasto con estado *por confirmar*, desde el punto de vista de un usuario que no pagó el gasto.

Notar que *Allie* no tiene permiso para editar el gasto, pero sí tiene la opción de confirmar el gasto. Si se hace click en *Confirmar*, se redirigirá al detalle del mismo gasto, pero esta vez puede verse que el estado del gasto cambió a *pagado*, ya que fue confirmado por todos los usuarios de la vivienda (Figura 79).

El gasto fue confirmado por todos los usuarios pertinentes.

El Depa

Detalle Gasto



Monto:	\$ 123
Creado por:	Levni
Usuario:	Levni
Categoría:	Teléfono
Foto:	ver foto
Fecha creacion:	8 de Junio de 2016
Fecha pago:	3 de Junio de 2016
Periodo:	2016-6
Estado:	pagado

Figura 79. Gasto aparece como *pagado* luego de ser confirmado por todos los usuarios de la vivienda.
Detalle del gasto desde el punto de vista de un usuario que no pagó.

En la siguiente sección se verá el detalle del mismo gasto, pero desde el punto de vista del usuario que pagó el gasto.

5.7.5. Modificar gasto

Un gasto *pendiente* puede ser modificado por cualquier usuario de la vivienda, mientras que un gasto con estado *por confirmar* o *pagado* sólo puede ser modificado por el usuario que pagó. Al modificar un gasto *pendiente* o *por confirmar*, éste mantiene su estado. Por otro lado, al modificar un gasto *pagado*, se revierte el estado a *por confirmar*. Esto se debe a que los usuarios que confirmaron dicho gasto pueden no estar de acuerdo con el mismo luego de las modificaciones.

De esta forma, cada usuario es responsable de sus contribuciones a la vivienda, ya que sólo el mismo usuario puede revertir el estado de sus gastos *pagados* a *por confirmar*. Al mismo tiempo, si un usuario ingresa un gasto sospechoso, sencillamente no será confirmado hasta que el usuario responsable realice las modificaciones exigidas por los compañeros.

A continuación, puede verse el gasto de la Figura 79, visto desde el punto de vista del usuario que pagó.

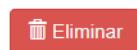


Monto:	\$ 123
Creado por:	Levni
Usuario:	Levni
Categoría:	Teléfono
Foto:	ver foto
Fecha creacion:	8 de Junio de 2016
Fecha pago:	3 de Junio de 2016
Periodo:	2016-6
Estado:	pagado

Figura 80. Detalle de gasto pagado por el mismo usuario.

Notar que, a diferencia del detalle que ven otros usuarios, el usuario que pagó puede ver el botón *Editar*. Es decir, a este usuario se le permite realizar modificaciones para esta instancia particular de gasto. Este botón redirecciona al usuario a la vista para editar el gasto (Figura 81).

Editar Gasto



Monto

Fecha pago

Figura 81. Vista para editar una instancia de gasto pagado.

Se utiliza el mismo *Datepicker* de *jquery-ui* para el ingreso de la fecha, visto al momento de crear nuevos gastos. Al enviar el formulario, se validan los valores ingresados siguiendo las mismas restricciones definidas para la creación de gastos, y si no existen errores, se redirige al usuario al detalle del mismo gasto, donde puede verse que el estado efectivamente cambió a *por confirmar* (Figura 82).

Gasto editado. Se cambió el estado a pendiente.

El Depa

Detalle Gasto

✎ Editarℹ

Monto:	\$ 456
Creado por:	Levni
Usuario:	Levni
Categoría:	Teléfono
Foto:	ver foto
Fecha creacion:	8 de Junio de 2016
Fecha pago:	3 de Junio de 2016
Periodo:	2016-6
Estado:	pendiente_confirmacion
Confirmado por:	Levni
Por confirmar:	Allie

Figura 82. Se modifica el monto del gasto, lo cual a su vez revierte el estado de *pagado* a *por confirmar*.

5.7.6. Eliminar gasto

Para la eliminación de un gasto se siguen las mismas restricciones que para la edición: cualquier usuario puede eliminar un gasto *pendiente*, pero sólo el usuario que pagó puede eliminar gastos con estado *por confirmar* o *pagado*. Como se vio en la Figura 81, el botón de *Eliminar* puede verse dentro de la vista para editar el gasto.

Al eliminar el gasto, se redirige al usuario a la lista de gastos. La siguiente figura muestra la interfaz que ve el usuario *Levni* al eliminar el gasto de la Figura 82.

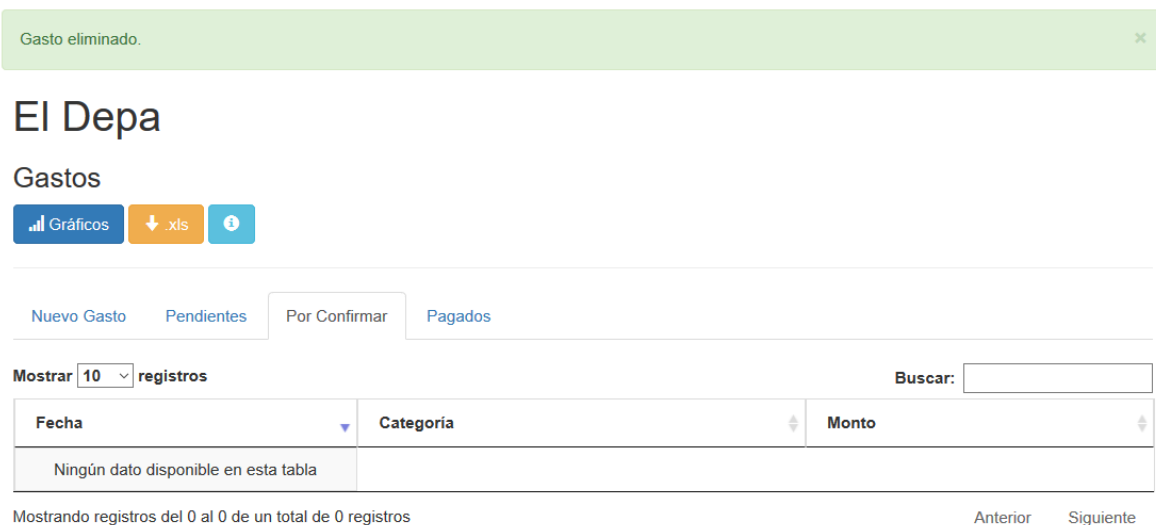


Figura 83. Luego de eliminar el gasto, se redirige al usuario a la lista de gastos.

Notar que el gasto no aparece en la lista de gastos con estado *por confirmar*. Los usuarios no pueden deshacer la acción de eliminar un gasto.

5.7.7. Graficar gastos

Es posible ver el historial de gastos de forma gráfica desde la página de configuración en *Gastos* -> *Gráficos*, o bien desde la barra de navegación en *Gastos*, y luego haciendo click en el botón *Gráficos*. La interfaz para generar gráficos puede verse en la siguiente figura:



Figura 84. Interfaz para generar los gráficos del historial de gastos.

El usuario puede definir tanto los períodos como las categorías que se desean graficar. Al hacer click en *Generar gráfico*, se mostrarán los totales invertidos en cada categoría seleccionada, para cada período seleccionado (Figura 85). Para efectos de la generación de gráficos, estos se cuentan independiente del usuario que pagó, pues se considera que el gasto es de la vivienda, y no de los usuarios individuales. Notar, sin embargo, que sólo se consideran los gastos con estado *pagado*.



Figura 85. Gráficos del total de gastos por categoría y por período.

Cabe mencionar dos librerías de javascript que hacen posible estas visualizaciones: *jquery-format*²⁵ y *Charts.js*²⁶. La primera consiste en una librería que facilita mostrar los números siguiendo el formato apropiado de la zona. Para el caso de Chile, esto quiere decir que las centenas estén separadas por puntos y los decimales por una coma (a diferencia de otros formatos donde es al revés).

Por otro lado, la segunda librería permite la generación de gráficos responsivos. Esta librería también es utilizada para la generación de los gráficos de los presupuestos y del balance, como se verá más adelante.

5.7.8. Exportar XLS

Es posible descargar el historial de gastos como hoja de cálculo de Excel (*.xls*) desde la interfaz que muestra el listado de gastos. Para esto, basta hacer click en el botón *.xls*, lo cual genera el archivo deseado (Figura 86).

²⁵ <http://www.asual.com/jquery/format/>

²⁶ <http://www.chartjs.org/docs/>

El Depa

Gastos



Nuevo Gasto Pendientes Por Confir

Categoría

Monto

Monto

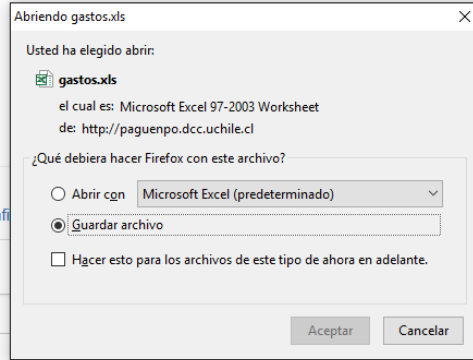


Figura 86. Descarga del historial de gastos en formato XLS.

El archivo en sí es muy similar a la tabla de la interfaz, ya que mantiene la misma separación de los gastos por estado (Figura 87).

	A	B	C	D	E	F	G
1	Fecha creación	Año pago	Mes pago	Día pago	Categoría	Pagado por	Monto
2	2016-06-08	2016	6	7	Internet	Levni	1234
3							
4							
5							
6							
7							
8							
9							
10							

Figura 87. Archivo XLS con el historial de gastos.

5.7.9. Sobre la privacidad

Por último, resulta importante mencionar que los gastos están asociados tanto a la vivienda como al usuario, y sólo son visibles para usuarios pertenecientes a la vivienda asociada al gasto. Si bien los gastos son accedidos a través de un parámetro en la URL²⁷, el cual claramente deja vulnerable el identificador de cada gasto en la base de datos en la misma URL (Figura 88), no es posible ingresar a cualquier gasto, pues el servidor revisa que se tengan los permisos necesarios antes de transformar dicho identificador en el objeto mismo de gasto.

²⁷ <https://docs.djangoproject.com/en/1.9/topics/http/urls/>

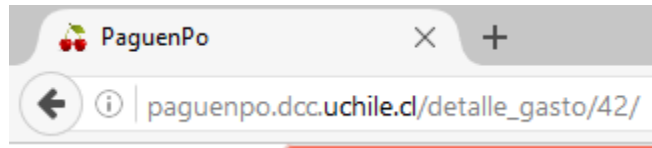


Figura 88. El detalle de los gastos se obtiene mediante el método GET.

Si un usuario de otra vivienda cambia el identificador en la URL con la intención de acceder a los gastos de otra vivienda, simplemente se le mostrará el siguiente mensaje de error:

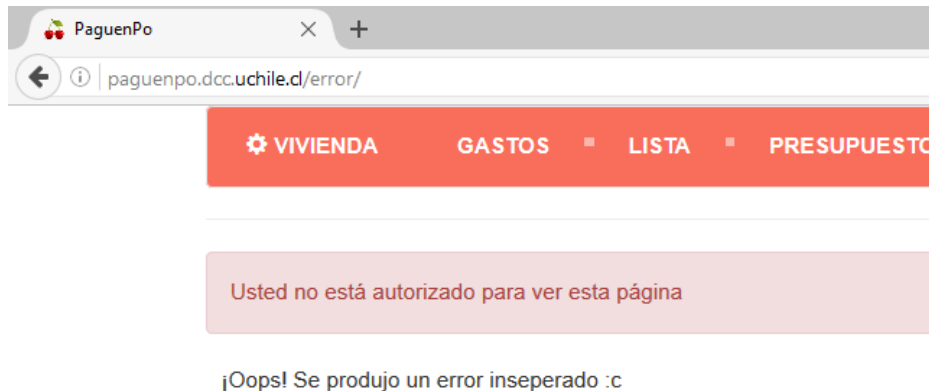


Figura 89. El usuario es redireccionado a la página de errores al intentar ingresar a un gasto externo.

Algo similar ocurre al intentar acceder a las fotos de los gastos ingresados por los usuarios, las cuales también se obtienen mediante identificadores en la URL. Este problema resulta bastante más complejo, y la solución es altamente dependiente del ambiente de producción que se tiene.

Actualmente se está corriendo el Django en el servidor Orion del DCC, en un *Virtualhost*²⁸ de Apache²⁹, donde, entre otras responsabilidades, es el Apache, y no Django, quien *sirve* los archivos (javascript, CSS e imágenes) al usuario. Esto presenta una dificultad, ya que el Apache no hace validaciones sobre las credenciales de los usuarios al momento de servir los archivos, lo que funciona muy bien para entregar los archivos *estáticos* necesarios (javascript y CSS), pero no así para servir *media* (fotos con restricciones).

Para solucionar esto, se guardaron las imágenes en un directorio distinto al del contenido estático (el cual es público), y se configuró el Apache para que restrinja el acceso a todas las solicitudes, excepto aquellas que contienen, entre otros, el header *X-Sendfile*. De esta forma, al intentar acceder a una imagen, el Django revisa las credenciales del usuario, y si efectivamente tiene

²⁸ <https://httpd.apache.org/docs/current/vhosts/>

²⁹ <http://www.apache.org/>

permitido acceder a la imagen, se agregan los headers adecuados a la solicitud, de tal forma que el Apache permita ver la imagen. En caso contrario, el apache niega el acceso.

5.8. Manejo de presupuestos

Cualquier usuario de la vivienda tiene permitido crear y editar los presupuestos definidos para las distintas categorías de la vivienda. Los presupuestos se crean para un período mes-año específico, y las visualizaciones y listados están organizados por este parámetro.

5.8.1. Listado de presupuestos

Los usuarios pueden acceder al listado de presupuestos (Figura 90) desde la página de configuración de la vivienda haciendo click en *Presupuestos -> Administrar*. El listado de presupuestos es una de las interfaces más sencillas, y sólo muestra los valores de los presupuestos para las categorías del período que se está consultando.



Figura 90. Listado de presupuestos.

Desde esta interfaz los usuarios pueden consultar los presupuestos de cualquier período haciendo click en *Anterior* o *Siguiente*. También pueden elegir ver esta información de forma gráfica, como se verá más adelante, además de crear nuevos presupuestos. Por último, al hacer click en el nombre de la categoría es posible modificar el valor del presupuesto para este período.

5.8.2. Crear presupuesto

Es posible crear un nuevo presupuesto desde el listado de presupuestos, desde la interfaz de gráficos de presupuestos, o bien desde la página de configuración de la vivienda, haciendo click en *Presupuestos -> Nuevo*. El formulario pertinente puede verse a continuación:

Nuevo Presupuesto

¿Reutilizar existente?

Categoría

Year month

Monto

Figura 91. Formulario para crear nuevo presupuesto.

Al crear el presupuesto, se redirige al usuario al listado de presupuestos, donde puede verse el nuevo elemento en la tabla, así como el mensaje indicando la creación del mismo (Figura 92).

El presupuesto fue creado exitosamente

El Depa

Presupuestos 2016-6

Mostrar registros

Categoría	Presupuesto
Gas	\$ 10.000
Internet	\$ 4.000

Mostrando registros del 1 al 2 de un total de 2 registros Anterior Siguiete

Figura 92. Se crea nuevo presupuesto para el período 2016-6.

5.8.3. Crear desde presupuesto existente

También es posible reutilizar los datos de un presupuesto ya creado para generar uno nuevo. Por ejemplo, los usuarios pueden querer repetir el presupuesto de *Gas* para el período siguiente. Para esto, basta hacer click en *Reutilizar existente* en el formulario de nuevo presupuesto (Figura 91). Esta acción despliega un nuevo formulario:

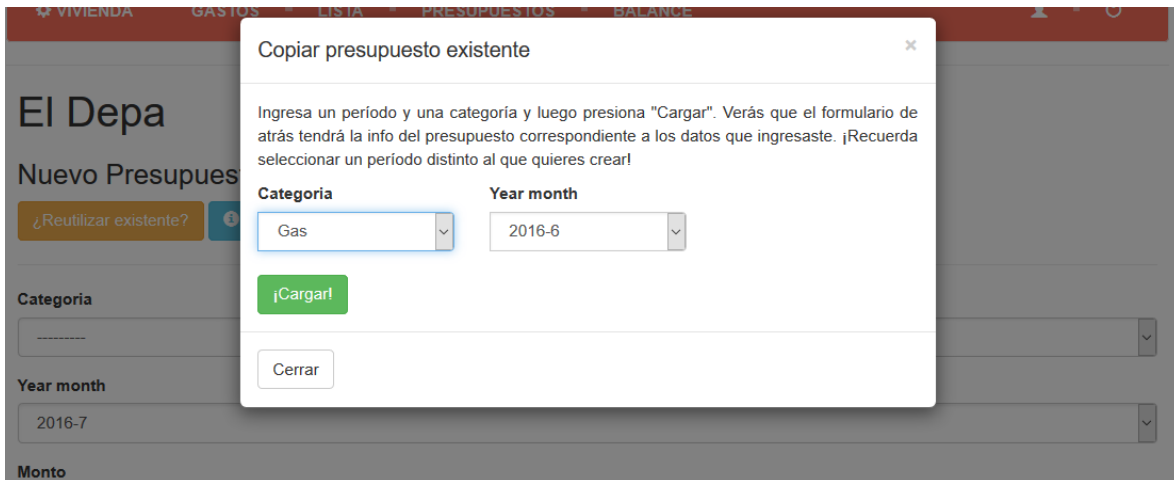


Figura 93. Formulario para reutilizar los datos de un presupuesto existente para crear uno nuevo.

La siguiente figura muestra como, al hacer click en *Cargar*, los datos del formulario original cambian:

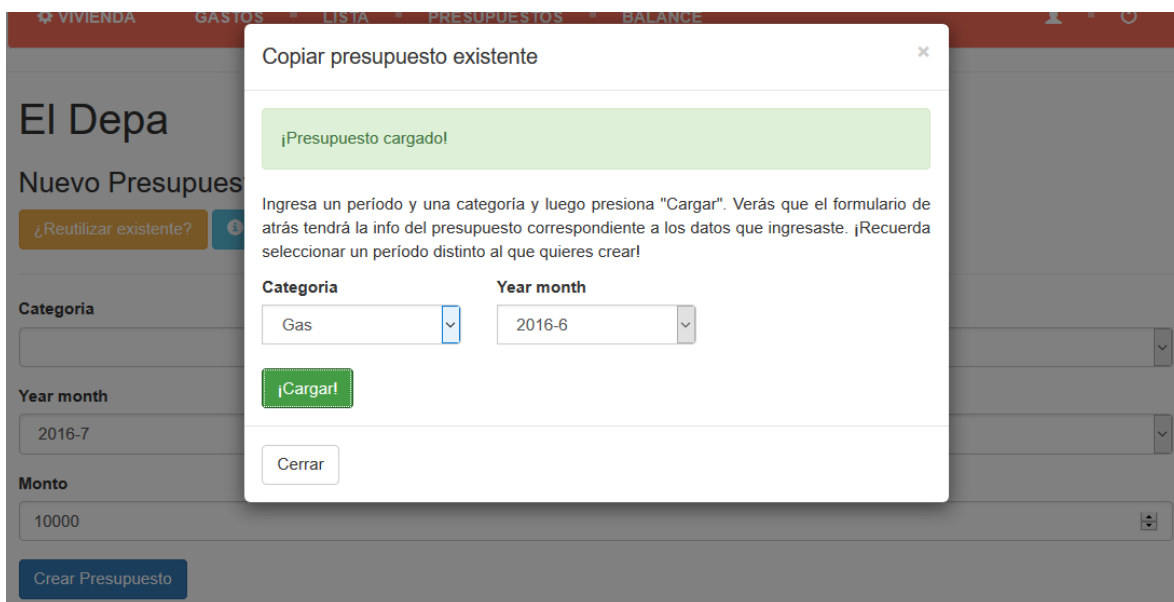


Figura 94. Al cargar los datos, el formulario de fondo cambia sus valores por los del presupuesto para *Gas* del período 2016-6.

Luego de esto, el usuario es capaz de crear el presupuesto de la misma forma que se vio en la sección anterior.

5.8.4. Modificar

Cualquier usuario de la vivienda puede modificar los presupuestos (Figura 95), para lo cual basta con hacer click en el nombre de la categoría del presupuesto, ya sea en la interfaz del listado de presupuestos, o de la interfaz de gráficos de presupuestos.

Presupuesto Gas período 2016 - 6



Monto

Figura 95. Formulario para modificar los valores de un presupuesto dado.

Al hacer click en *Actualizar monto* se cambia el valor del presupuesto, y se redirige al listado de presupuestos.

5.8.5. Gráficos

Esta interfaz es la que más información entrega acerca de los presupuestos. Se puede acceder desde la barra de navegación, o bien desde la página de configuración, en *Presupuestos -> Gráficos*.

Presupuestos 2016-6

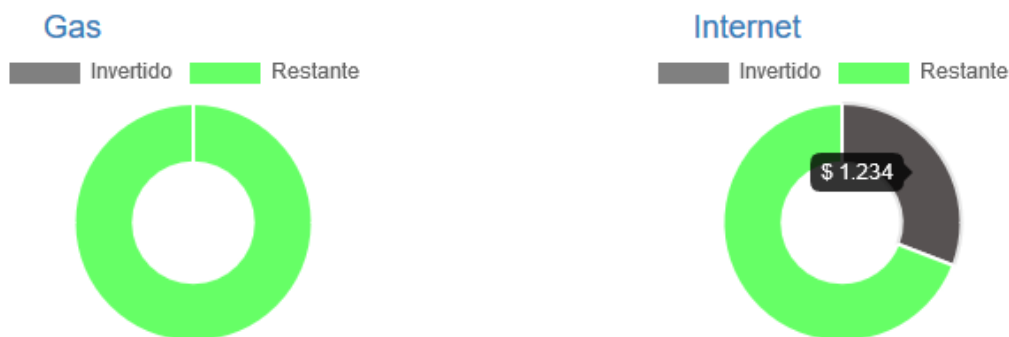


Figura 96. Interfaz que muestra gráficamente los presupuestos definidos para un período dado. Al colocar el puntero sobre las distintas partes del gráfico puede verse el detalle.

Como puede verse en la figura, esta interfaz permite la misma navegación que el listado de presupuestos: *Siguiente*, *Anterior* y *Nuevo*, así como un link al resumen o listado de presupuestos. Esta interfaz tiene la ventaja adicional de mostrar qué porcentaje del presupuesto se ha alcanzado

hasta el momento. Para esto, el servidor hace la suma de todos los gastos que fueron pagados en el período dado y para la categoría de cada presupuesto. Es importante notar que sólo se cuentan los gastos que ya fueron confirmados por todos los usuarios, y por lo tanto tiene estado *pagado*. Los gráficos se generan utilizando las mismas librerías que se vieron para la visualización de gastos.

5.9. Manejo de ítems

Los ítems facilitan la generación de listas de compras, al acelerar el proceso de ingreso de ítems que suelen repetirse en muchas listas. Una vivienda tiene asociado un conjunto de ítems, y al crear una nueva vivienda, esta viene con un conjunto predefinido de ítems, los cuales apuntan a reducir el número de configuraciones que deben hacer los usuarios antes de comenzar a utilizar listas de compras.

Luego, los usuarios pueden crear nuevos ítems que se ajustan a las necesidades propias de la vivienda, e incluso editar los valores de los mismos. De esta forma, los usuarios tienen permitido medir las unidades de medida como mejor les parezca y mejor se entiendan entre ellos.

Resulta muy importante notar que, si bien los ítems se utilizan para generar listas, la lista recibe sólo una *referencia* al ítem, por lo que, al eliminar un ítem de la lista, éste sigue perteneciendo al conjunto de ítems de la vivienda, y puede ser utilizado en la generación de otra lista totalmente distinta más adelante.

5.9.1. Listado

Se puede ver el listado de Ítems (Figura 97) desde la página de configuración en *Ítems -> Administrar*.

Items

+ Nuevo ⓘ

Mostrar 10 registros Buscar:

Nombre	Unidad	Descripción
Aceite	botella	
Aceite de Oliva	botella	
Aceto balsámico	botella	
Ajo	unidad	
Ajo sal	unidad	
Arroz	paquete	
Azúcar	kg	
Cabellitos de ángel	kg	
Café de grano	paquete	
Café instantaneo	tarro	

Mostrando registros del 1 al 10 de un total de 79 registros Anterior 2 3 4 5 ... 8 Siguiente

Figura 97. Listado de ítems de la vivienda.

Desde esta interfaz se tiene acceso tanto a la creación de nuevos ítems como a la edición de ítems existentes (haciendo click en el nombre del ítem).

5.9.2. Crear

Se puede ver el formulario de creación de ítems (Figura 98) al hacer click en el botón *Nuevo* presente en el listado de ítems, o bien desde la página de configuración en *Ítems -> Nuevo*.

Nuevo Ítem

Nombre

Unidad medida


Descripcion

Figura 98. Formulario para crear nuevo ítem.

Notar que la unidad de medida corresponde a un campo de texto en lugar de un *select*, ya que los usuarios pueden tener sus propias formas de medir los ítems. Se permite agregar además una descripción del ítem, como puede ser, por ejemplo, la marca preferida por la mayoría de los usuarios. Esta descripción podrá ser vista más adelante, cuando un usuario quiera completar una lista.

Al llenar el formulario y hacer click en *Crear*, se redirige al usuario al listado de ítems, donde puede verse el nuevo ítem (Figura 99).

Items

+ Nuevo 

Mostrar registros Buscar:

Nombre	Unidad	Descripción
itemDeEjemplo	unidad	alguna descripción adicional del ítem

Mostrando registros del 1 al 1 de un total de 1 registros (filtrado de un total de 80 registros) Anterior Siguiente

Figura 99. Nuevo ítem aparece en el listado de ítems de la vivienda. Notar el uso del filtro de la tabla para facilitar encontrar el ítem deseado.

5.9.3. Editar

Al hacer click en el nombre del ítem en el listado, se puede ver el formulario para editar dicho ítem. A continuación, se muestra dicho formulario para el ítem creado en la sección anterior:

Nombre

Unidad medida

Descripcion

Figura 100. Formulario de edición de ítem.

Si se cambia algún campo y se hace click en *Editar*, y sólo luego de verificar que los valores ingresados sean válidos, se redirige al usuario al listado de ítems, donde pueden verse los nuevos valores (Figura 101).

Items



Mostrar registros Buscar:

Nombre	Unidad	Descripción
itemDeEjemplo	tarro	otra descripción

Mostrando registros del 1 al 1 de un total de 1 registros (filtrado de un total de 80 registros) Anterior Siguiente

Figura 101. Se edita el ítem y pueden verse los nuevos valores.

Notar que tanto la descripción como la unidad de medida fueron cambiados.

5.10. Manejo de lista de compras

Cada vivienda puede tener asociada a lo más 1 lista de compras pendiente. Vale decir, si se tiene una lista de compras con estado pendiente, los usuarios no pueden crear una nueva lista, sino editar la lista actual (cambiar los ítems pedidos) o bien completar la lista. A continuación, se describe el manejo básico de las listas de compras: creación y edición. En la sección siguiente se verá el proceso de completar la lista de compras pendiente.

5.10.1. Crear lista de compras

Si no existe una lista pendiente, los usuarios pueden crear una nueva lista haciendo click en el botón *Lista* de la barra de navegación, lo cual los llevará al formulario para crear la lista (Figura 102).

Nueva Lista



Ítem	Cantidad	
<input type="button" value="+ Agregar ítem"/>		

Figura 102. Formulario inicial para crear una nueva lista.

Al hacer click en el botón *Agregar Ítem* van apareciendo nuevas filas en la tabla (Figura 103). Cada fila corresponde al formulario individual para agregar 1 nuevo ítem a la lista.

Nueva Lista



ítem	Cantidad	
+ Agregar ítem		
<input type="text" value="ítem"/>	<input type="text" value="Cantidad"/>	
<input type="text" value="ítem"/>	<input type="text" value="Cantidad"/>	

✓ Generar lista

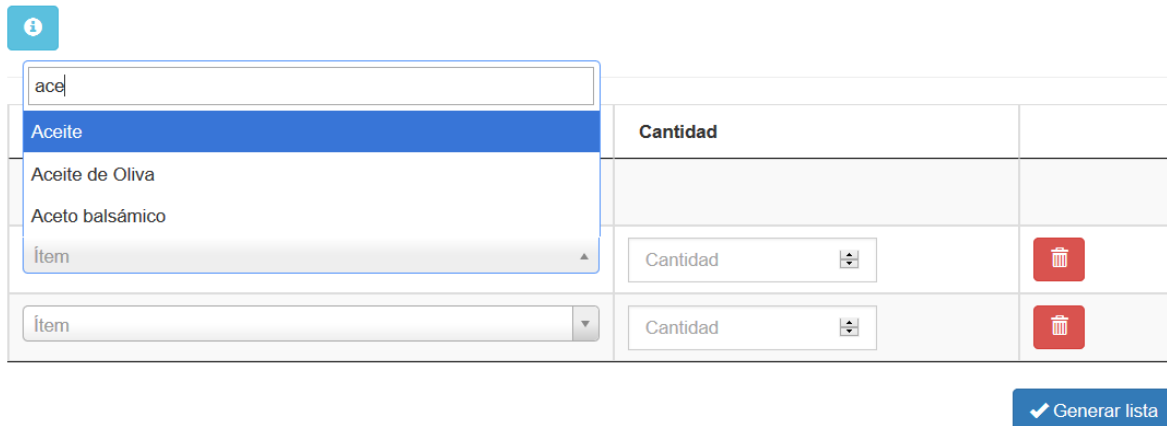
Figura 103. Formulario para lista con 2 ítems.

Las dos primeras columnas corresponden al nombre del ítem y a la cantidad solicitada, respectivamente. La tercera columna contiene un ícono, el cual elimina la fila al hacerle click. Para ingresar el nombre del ítem, el usuario debe seleccionar un ítem de la lista que se le provee. Estos ítems corresponden a aquellos que se encuentran definidos para la vivienda, como se vio en la sección anterior. Notar que el usuario no tiene permitido ingresar un ítem que no se encuentra en el listado; para esto, se debe crear primero el ítem en la interfaz apropiada, y luego hacer click en *Agregar Ítem*.

Para facilitar el ingreso del ítem, se hace uso de un plugin de jQuery llamado *Select2*³⁰, el cual mejora la usabilidad del listado al permitir ver todos los ítems (al igual que un campo *select* normal), y al mismo tiempo filtrar la lista con un campo de texto (Figura 104).

³⁰ <https://select2.github.io/>

Nueva Lista



The screenshot shows a form titled "Nueva Lista". At the top left, there is a blue information icon. Below it is a search input field containing the text "ace". A dropdown menu is open below the search field, listing "Aceite", "Aceite de Oliva", and "Aceto balsámico". Below the dropdown is a table with two columns: "Ítem" and "Cantidad". The table has two rows. The first row has "Aceite" in the "Ítem" column and "Cantidad" in the "Cantidad" column. The second row has "Ítem" in the "Ítem" column and "Cantidad" in the "Cantidad" column. To the right of the table, there are two red trash icons. At the bottom right, there is a blue button with a checkmark and the text "Generar lista".

Figura 104. Select2 permite filtrar el listado de ítems.

Al seleccionar un ítem, puede verse que el *placeholder* del campo de *cantidad* correspondiente cambia para mostrar la unidad de medida en que deben solicitarse los ítems (Figura 105).



The screenshot shows the "Nueva Lista" form with the "Aceite" item selected in the "Ítem" dropdown. The "Cantidad" dropdown now shows "botella" as the selected unit of measurement. The "Ítem" dropdown still shows "Ítem" as the selected item. The "Cantidad" dropdown has a red trash icon to its right.

Figura 105. Al seleccionar un ítem, se le indica al usuario la unidad de medida apropiada, en este caso "botella".

Al hacer click en *Generar lista* se le pide al usuario que confirme los ítems ingresados (Figura 106), para luego crear la lista y redirigirlo al formulario para completar dicha lista.

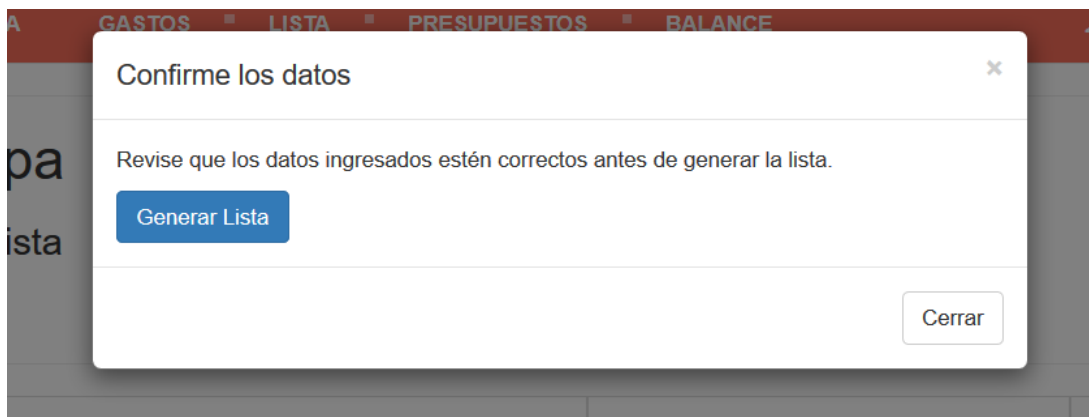


Figura 106. El usuario debe confirmar los datos antes de crear la lista.

Al momento de crear la lista, se tienen algunas restricciones: el ítem ingresado debe existir en la base de datos, y la cantidad solicitada debe ser un número entero positivo. Además, no pueden repetirse los ítems en la lista. Si se tienen ítems repetidos, se redirige a la misma página, indicando el error. Por otro lado, las filas que contienen errores son ignoradas. Si no se ingresa ninguna fila válida, la lista no se crea, y se le indica el error al usuario.




5.10.2. Modificar lista de compras

Una vez que se tiene una lista de compras creada, los usuarios pueden completarla (Figura 107) haciendo click en el botón *Lista* de la barra de navegación³¹.

³¹ Notar que es la misma forma de acceder al formulario para crear una nueva lista; al momento de generar la barra de navegación, se verifica si existe o no una lista de compras pendiente, y se crea el link dependiendo de esta condición. De esta forma, el mismo botón puede redirigir al usuario a distintas interfaces.

Lista 15 de Junio de 2016 a las 18:55



Ítem	Cant.	Esta Compra
Pan (unidad)	2	 + <input type="text"/>  -
Aceite (botella)	3	 + <input type="text"/>  -

¿Qué hacer con ítems restantes?



¿Subir foto de la boleta?



Monto total	
-------------	--

Figura 107. Formulario para completar lista.

Este formulario se verá en más detalle en la sección siguiente, en la que se muestra cómo completar una lista. Lo que hay que notar en esta sección es el botón *Editar*, el cual lleva al usuario a la interfaz para editar la lista de compras (Figura 108).

Editar Lista



Ítem	Cantidad	Eliminar
<input type="text" value="Aceite (botella)"/>	<input type="text" value="3"/>	<input type="checkbox"/> Eliminar
<input type="text" value="Pan (unidad)"/>	<input type="text" value="2"/>	<input type="checkbox"/> Eliminar



Figura 108. Formulario para editar una lista de compras.

La primera diferencia que salta a la vista corresponde a la tercera columna. El formulario para crear listas permitía eliminar fácilmente las filas no deseadas, mientras que el formulario de edición exige marcar las filas a eliminar, en lugar de borrarlas directamente. Esto ocurre debido a que el formulario para crear está definiendo los atributos de un objeto que aún no existe en la base de datos, por lo que el acto de agregar y luego borrar una fila no afecta a la base de datos: sencillamente nunca se guarda dicha fila.

Por otro lado, el formulario de edición trabaja con un objeto que ya existe y que se está modificando, por lo que el acto de borrar una fila modifica directamente la base de datos. De la forma en que se presenta el formulario, el usuario puede arrepentirse de eliminar una fila *antes* de guardar los cambios, cosa que no podría hacerse si la fila desapareciese.

El resto del formulario funciona de manera similar al de creación: el usuario debe elegir un ítem de la lista, la cual hace uso de *Select2*, para luego indicar la cantidad exigida. Al momento de guardar los cambios se imponen las mismas condiciones de creación de listas: cantidades enteras positivas e ítems existentes en la base de datos (Figura 109).

Ítem	Cantidad	Eliminar
Aceite (botella)	-3 <small>Asegúrese de que este valor es mayor o igual a 0.</small>	<input type="checkbox"/> Eliminar
Pan (unidad)	2	<input type="checkbox"/> Eliminar
Pan (unidad)	4	<input type="checkbox"/> Eliminar
<div style="background-color: #f8d7da; padding: 5px; border: 1px solid #f5c6cb;"> Ítem lista con este Ítem y Lista ya existe. × </div>		

Figura 109. El usuario intenta modificar la lista con campos inválidos, por lo que la lista no se guarda, y se le muestran los errores.

Al hacer click en *Guardar Cambios*, se le pide al usuario que confirme, y luego, si no hay errores, se redirige al formulario para completar la lista.

5.11. Completar una lista de compras

Una vez que se tiene creada la lista, los usuarios pueden completarla, marcando así los ítems de la lista como *comprados*, y al mismo tiempo generando un gasto asociado a la lista. La interfaz para completar corresponde a la que se vio brevemente en la Figura 107, sin embargo, esta interfaz está diseñada especialmente para ser utilizada desde el smartphone, ya que corresponde justamente al formulario que los usuarios completan desde el supermercado.

Debido a lo anterior, las figuras de esta sección serán vistas desde la perspectiva de un usuario que usa la aplicación desde su smartphone. Es importante recordar, sin embargo, que todas las interfaces se adaptan al dispositivo y se ven relativamente bien en una pantalla pequeña, pero no fueron diseñadas específicamente para dicho uso. De esta forma, la interfaz con la que realmente trabajarán los usuarios corresponde a la siguiente:

15° 4G 19:22

El Depa

Lista 15 de Junio de 2016 a las 18:55

Editar i

Ítem	Cant.	Esta Compra
Pan (unidad)	2	⚡ + <input type="text"/> -
Aceite (botella)	3	⚡ + <input type="text"/> -

¿Qué hacer con ítems restantes?

Nueva lista Descartar

¿Subir foto de la boleta?

Foto

Monto total Pagar

Figura 110. Formulario para completar lista de compras visto desde un smartphone *moto G 2015*.

Para completar la lista, los usuarios pueden hacer click en el campo numérico, o bien hacer click en los botones + y - para variar la cantidad comprada. Esta última opción resulta atractiva ya que para rellenar los campos manualmente, es necesario que aparezca el teclado numérico del smartphone, lo que oculta gran parte del formulario, incluyendo los siguientes elementos de la lista.

El ícono del rayo tiene la función de rellenar el valor comprado con el valor solicitado en 1 sólo click, lo que resulta aún más atractivo que los botones para aumentar/disminuir la cantidad comprada. Vale decir, si se hace click en el rayo de la fila *Pan*, se marcará que se compraron 2, mientras que el rayo de la fila *Aceite* marcará un 3.

El usuario tiene permitido comprar sólo parcialmente la lista, dejando algunos campos en blanco. De esta forma, luego de ingresar las cantidades compradas de cada ítem, se le permite al usuario crear una nueva lista automáticamente con los ítems que no marcó como comprados (Figura 111), o bien descartar dichos ítems.

Por último, se le pide al usuario que ingrese una foto de la boleta e ingrese el monto total de la compra. Estos campos se utilizan para crear el gasto asociado a la compra de la lista, y se comportan de la misma forma que los que se vieron en la sección de Gastos.



Figura 111. El usuario puede crear una nueva lista con los ítems que no se compraron en esta instancia.

Al hacer click en *Pagar*, se pide que revise los datos ingresados, para luego redirigir al detalle del gasto asociado a la lista (Figura 112).

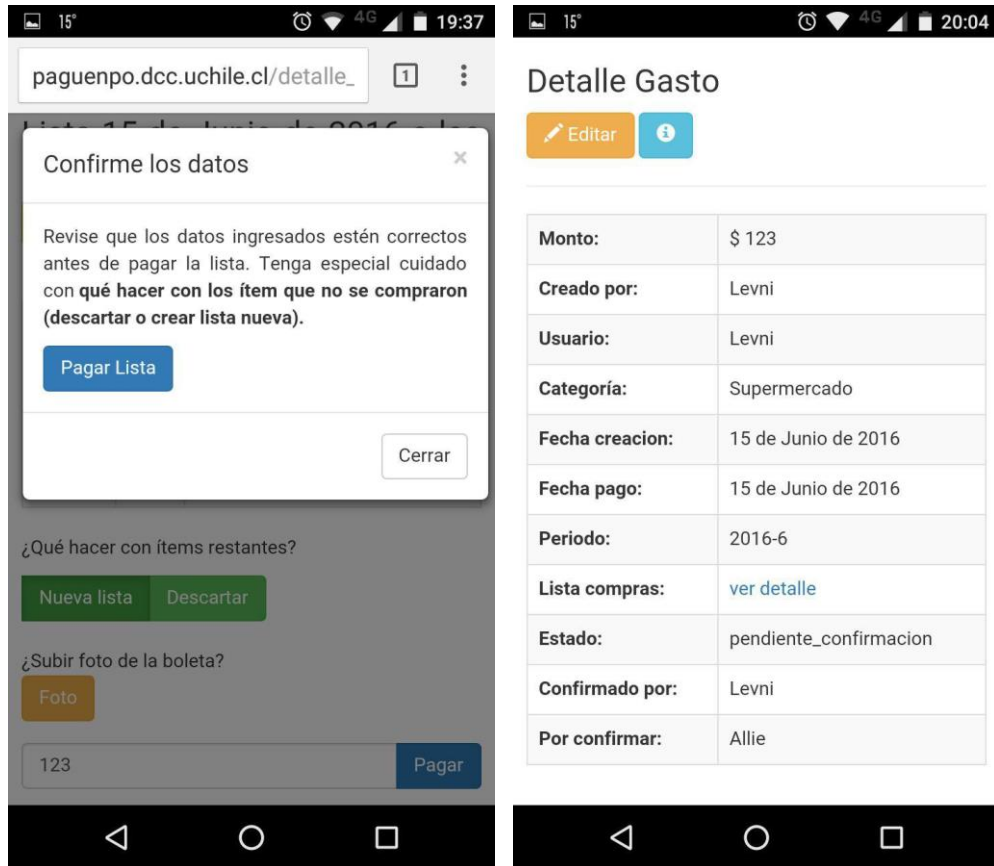


Figura 112. *Izquierda:* Se pide confirmación del usuario antes de marcar los ítems indicados como comprados. *Derecha:* se redirige al usuario al detalle del gasto asociado a la lista pagada.

El gasto que se genera de forma automática se comporta igual que los gastos creados de forma directa, por lo que debe ser confirmado por los demás usuarios antes de ser considerado como *pagado*. A diferencia de los gastos creados de forma directa, la figura anterior muestra un link al detalle de la lista de compras asociada al gasto. Este detalle se ve de la siguiente forma:



Figura 113. Detalle de la lista de compras pagada.

Al hacer click en el monto total se redirige al usuario al detalle del gasto asociado, el cual corresponde al visto anteriormente. Finalmente, recordemos que al pagar la lista de compras de la Figura 111, se eligió crear una nueva lista usando los ítems no comprados. Si se ingresa a la lista de la vivienda, se puede ver que esta es la nueva lista pendiente de la vivienda:



Figura 114. Lista de compras creada a partir de los ítems que no fueron comprados en una lista anterior.

Finalmente, cabe mencionar que, si el usuario elige descartar los ítems no comprados, estos sencillamente se eliminan, y no se crea una nueva lista. Luego, si se hace click en *Lista* en la barra de navegación, los usuarios verán el formulario para crear una nueva lista de compras, pues no existirá ninguna lista pendiente.

5.12. Manejo de vacaciones

Los usuarios pueden definir períodos de vacaciones, lo que puede resultar útil si alguno se encontrase fuera de la vivienda por un tiempo prolongado, pero al mismo tiempo se quiere tener un registro verídico del balance de gastos de la vivienda. Esto quiere decir que, al momento de calcular los balances de gastos, el sistema tiene en cuenta los períodos en que los usuarios no estuvieron presentes, y por lo tanto no se les deben cobrar los gastos realizados durante dicho período. Esto genera un balance más cercano a la realidad de la vivienda, y minimiza los ajustes manuales que se deben hacer a partir del cálculo automático del balance de gastos. El balance de gastos también toma en cuenta otros factores, los cuales se verán más adelante. En las siguientes subsecciones se verán las interfaces para crear y modificar instancias de vacaciones.

5.12.1. Crear vacación

El formulario para crear una nueva instancia de vacación (Figura 115) puede ser accedido desde la página de configuración haciendo click en *Vacaciones -> Administrar -> Nuevo*, o bien directamente haciendo click en *Vacaciones -> Nueva*.

Nueva Vacación

Fecha inicio

Fecha fin

Figura 115. Formulario para crear nueva instancia de vacación.

La vacación consiste simplemente en un intervalo de tiempo. Para el ingreso de fechas, al igual que en el caso de los gastos pagados, se hace uso del *Datepicker* de jquery-ui (Figura 116).



Figura 116. Se utiliza jquery-ui para facilitar el correcto ingreso de las fechas.

Naturalmente, la fecha final de la vacación debe ser posterior a la fecha inicial. El servidor hace esta validación, e indica el error de forma apropiada (Figura 117).

La fecha final debe ser posterior a la fecha inicial.

El Depa

Nueva Vacación

Fecha inicio

2016-06-18

Fecha fin

2016-06-12

Crear

Figura 117. Se valida que la fecha final sea posterior a la inicial.

Si se ingresa un intervalo válido de fechas, se redirige al usuario al listado de vacaciones, donde puede verse la nueva instancia de vacación (Figura 118).

¡Salida creada correctamente!

El Depa

Vacaciones

+ Nueva



Mostrar 10 registros

Buscar:

Usuario	Inicio	Fin
Levni	23 de Junio de 2016	30 de Junio de 2016

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior

1

Siguiente

Figura 118. Listado de vacaciones definidas para los usuarios de la vivienda, donde puede verse la vacación que se acaba de crear.

Si se hace click en el nombre del usuario puede verse el detalle de la vacación, y editar el intervalo.

5.12.2. Modificar vacación

Para modificar el intervalo de una instancia de vacación, el usuario debe ingresar al listado de vacaciones desde *Vacaciones* -> *Administrar* en la página de configuración, y luego hacer click en la vacación que se desea editar. Notar que la vacación sólo puede ser editada por el usuario que la creó; si un usuario distinto intenta editarla, será redirigido a una página de error, y no se le mostrará el formulario de edición (Figura 119).

Editar Vacación

Fecha inicio

Fecha fin

Figura 119. Formulario de edición de vacaciones.

Este formulario tiene las mismas restricciones que el formulario de creación, y también hace uso de jquery-ui para el ingreso de fechas (Figura 120).

Editar Vacación

Fecha inicio

Fecha fin

Figura 120. Se cambia la fecha de fin de la vacación.

Al hacer click en *Actualizar* se validan los datos ingresados, y de no haber errores se redirige al usuario al listado de vacaciones, donde pueden verse los valores actualizados (Figura 121).

¡Vacación actualizada!

El Depa

Vacaciones

+ Nueva 

Mostrar registros

Buscar:

Usuario	Inicio	Fin
Levni	23 de Junio de 2016	2 de Julio de 2016

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior Siguiente

Figura 121. Los nuevos valores pueden verse en el listado de vacaciones.

5.13. Sobre el balance

El balance de gastos corresponde a la diferencia entre el total que cada usuario ha invertido en la vivienda y el total de lo que deberían haber invertido.

5.13.1. Visualizar balance

Para acceder a la visualización del balance de gastos de la vivienda (Figura 122), basta hacer click en *Balance* en la barra de navegación, o bien desde la página de configuración en *Gastos -> Balance*.

Balance de Gastos

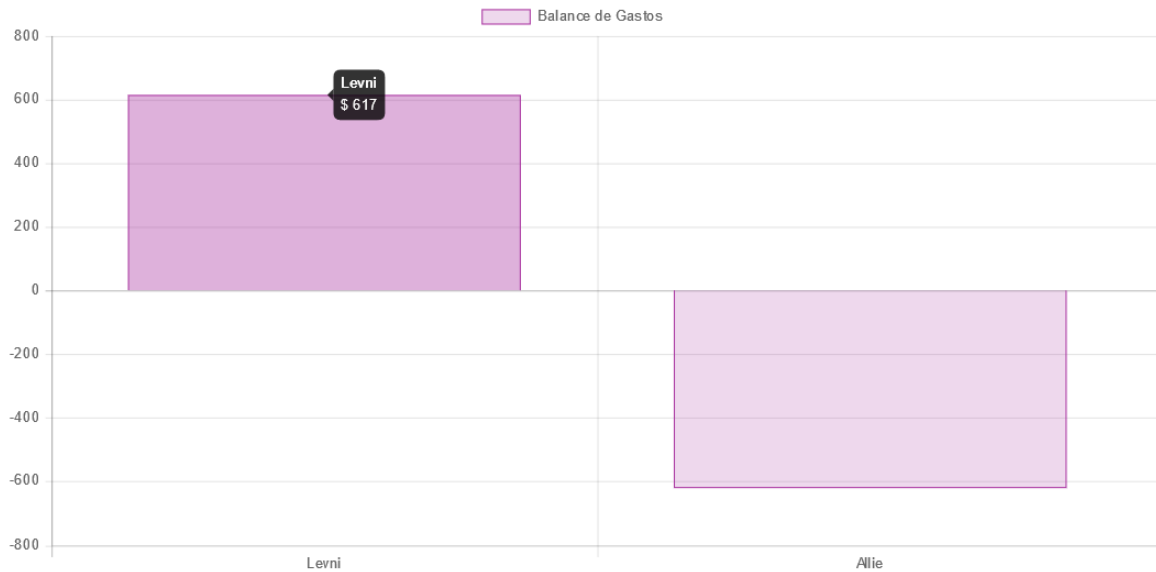


Figura 122. Representación gráfica del balance de gastos.

El gráfico de la figura muestra al usuario *Levni* con un balance positivo, mientras que *Allie* tiene un balance negativo. Esto quiere decir que *Levni* ha invertido más de lo que debería, mientras que *Allie* ha gastado menos de lo que debería.

Esto no quiere decir que *Allie* no haya invertido nada, o mucho menos, que haya invertido un monto negativo. El gráfico simplemente muestra la diferencia entre lo que *debería* haber invertido cada usuario, y lo que efectivamente se invirtió. De esta forma, el gráfico no indica cuánto ha gastado cada usuario, sino qué tan desbalanceadas se encuentran las contribuciones.

Si se hace click en el botón *¿Cómo balancear?* los pueden ver un listado con instrucciones de transferencias (Figura 123) que llevarían a la vivienda a un estado balanceado.



Para balancear los gastos de todos los usuarios, se deben seguir estas instrucciones:

- Allie debe transferir:
 - \$617 a Levni

Figura 123. Instrucciones para balancear.

Luego, los usuarios pueden hacer las *transferencias* necesarias de tal forma que todos queden balanceados. Esto se verá en la sección siguiente. Por otro lado, en la siguiente subsección se verá el detalle del cálculo del balance.

5.13.2. Sobre el cálculo de los balances

Como ya se dijo, el balance de gastos corresponde a la diferencia entre el total *esperado* y el total *real* que cada usuario ha invertido en la vivienda.

El total *esperado* corresponde a lo que el usuario debería haber invertido. Para calcular este valor, se revisa cada gasto de la vivienda, y se identifica a los usuarios *responsables* o *afectados* por el gasto. Vale decir, dado un gasto, se identifica la fecha del mismo, y luego se descarta a los usuarios que no pertenecían a la vivienda en dicha fecha.

Seguidamente, se identifica si la categoría del gasto es compartida en vacación. Si esto no es así, se descarta a los usuarios que se encontraban de vacación en la fecha del gasto (ya que no les corresponde pagar el gasto). Por otro lado, de ser compartido en vacación, no se descarta a ningún usuario, independiente de los períodos de vacación definidos.

Finalmente, el total del gasto se divide equitativamente entre el conjunto de usuarios responsables, incrementando así el total esperado de cada usuario.

El total *real* de cada usuario corresponde a la suma de los gastos pagados por el mismo que ya fueron confirmados. Sin embargo, sólo se cuenta la porción del gasto que es *percibida* por los usuarios actuales. Esto se hace para considerar varios casos de borde. El siguiente ejemplo muestra por qué esta diferenciación es importante:

- Se tiene una vivienda con usuario A, B y C
- Usuario A realiza gasto por un monto X
- Usuario C abandona la vivienda
- Usuario D ingresa a la vivienda

En este caso, el total invertido por A no corresponde a la totalidad de X, ya que la porción *percibida por la vivienda actual* corresponde a $2/3$ de X, pues sólo fue percibido por 2 de los 3 usuarios activos actualmente. En este ejemplo, se tendría la siguiente tabla de balance:

Tabla 3. Tabla de totales para los usuarios del ejemplo 1.

Usuario	Real	Esperado	Balance
A	$2/3 X$	$1/3 X$	$+1/3 X$
B	0	$1/3 X$	$-1/3 X$
D	0	0	0

De esta forma, los usuarios A y B están pagando la porción que les correspondía originalmente ($1/3$), y a D no se le está cobrando nada, pues no pertenecía a la vivienda al momento del gasto. Notar que, de esta forma, la suma de los balances siempre suma 0, lo que tiene sentido ya que de otra forma no sería posible generar el listado de instrucciones para balancear, las cuales se deducen de la columna de balance: B debe transferir $1/3$ de X a A.

Sin embargo, la realidad es que A *sí* pagó la totalidad de X, por lo tanto, resulta normal preguntarse dónde está el tercio restante que no se le ha pagado. Esto se explica debido a que se está haciendo la siguiente suposición: cuando C abandonó la vivienda, éste pagó todas sus deudas con sus compañeros, ya sea usando el sistema o no.

Es de vital importancia hacer esta suposición, ya que no tendría sentido seguir llevando la cuenta del balance de un usuario que no se encuentra en la vivienda, el cual por lo tanto no tiene la capacidad de realizar transferencias a los usuarios que aún pertenecen a ésta.

En un sistema perfecto donde se tiene registro de absolutamente todas las transacciones, podría resultar erróneo hacer esta suposición, pero es necesario tener flexibilidad, ya que resulta poco probable que el usuario que abandona la vivienda efectivamente se dé el tiempo de hacer las transferencias necesarias en el sistema antes de irse.

Un caso muy similar es aquel en que un usuario, en vez de abandonar la vivienda, simplemente se encuentra de vacaciones:

- Se tiene una vivienda con usuario A, B y C
- Usuario C se encuentra de vacaciones
- Usuario A realiza gasto por un monto X el cual *no* es compartido en vacaciones

En este caso, se tendría la siguiente tabla:

Tabla 4. Tabla de totales para los usuarios del ejemplo 2.

Usuario	Real	Esperado	Balance
A	X	$1/2 X$	$+1/2 X$
B	0	$1/2 X$	$-1/2 X$
C	0	0	0

A diferencia del ejemplo anterior, se tiene que el gasto se divide en 2, ya que los usuarios que percibieron el gasto fueron A y B. en este caso, se considera que A efectivamente pagó el total del gasto, siendo que le correspondía la mitad. Claramente es B quien debe reembolsarle la mitad del gasto, pues este estaba presente al momento del gasto. Por último, hace sentido que al usuario C no se le cobre nada, pues se encontraba de vacaciones al momento del gasto. Por último, cabe mencionar el siguiente ejemplo:

- Se tiene una vivienda con usuario A, B y C
- Usuario A realiza gasto por un monto X
- Usuarios B y C abandonan la vivienda
- Usuario D ingresa a la vivienda

En este caso, se tendría la siguiente tabla:

Tabla 5. Tabla de totales para los usuarios del ejemplo 3.

Usuario	Real	Esperado	Balance
A	$1/3 X$	$1/3 X$	0
D	0	0	0

En este caso, el balance de A corresponde a 0, ya que, como ya se vio, se asume que los usuarios que abandonaron la vivienda saldaron sus deudas. De esta forma, el usuario D no ve ninguna parte del gasto X, lo que resulta coherente, ya que realmente no lo percibió.

Con estos 3 casos considerados, es posible resolver sistemas mucho más complejos con variados números de vacaciones por usuario, donde se tiene siempre el invariante de que la suma de los balances debe dar siempre 0.

5.14. Manejo de transferencias

Las transferencias corresponden al mecanismo que tienen los usuarios para indicarle al sistema que se llevaron a cabo ajustes de cuentas, de tal forma de llevar un balance de gastos más fiel a la realidad de la vivienda.

Existen 2 interfaces para realizar transferencias: puede hacerse desde la interfaz de balance haciendo click en *Transferir*, lo cual despliega el formulario de transferencias (Figura 124), o bien directamente desde la página de configuración, haciendo click en *Usuarios -> Transferencias* (Figura 125).

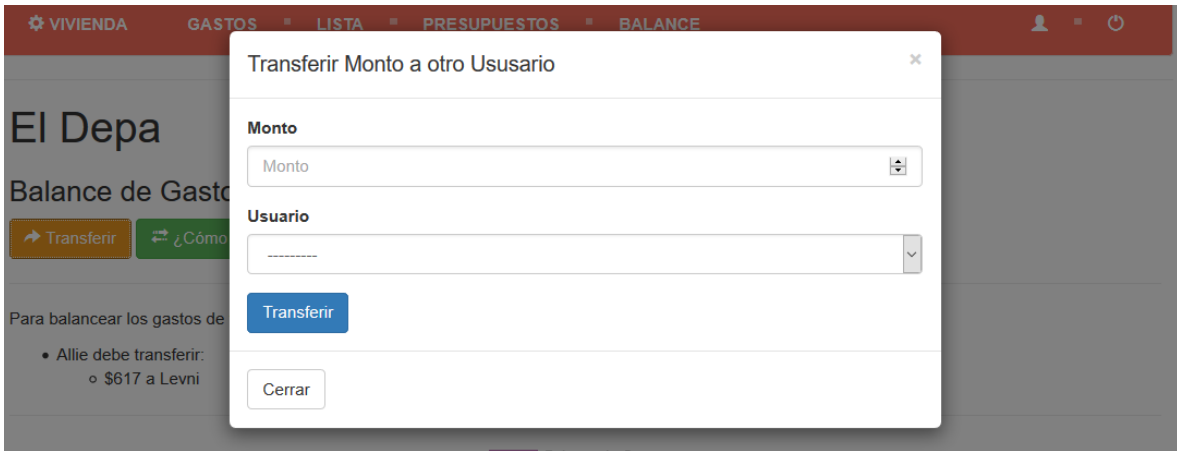


Figura 124. Formulario para realizar transferencias visto desde el balance de gastos.

Transferencias



Monto

Usuario

Transferir

Figura 125. Formulario de transferencias.

El formulario es el mismo, por lo que se trabajará con el de la Figura 125. El usuario debe ingresar a cuánto se desea transferir, y a cuál usuario.

Transferencias



Monto

Usuario

Transferir

Figura 126. Usuario selecciona un monto y usuario para transferir. Notar que la transferencia corresponde a la que sugirió el sistema en la Figura 123.

En la figura anterior, se tiene al usuario *Allie*, quien realizará una transferencia a su compañero *Levni* por el monto sugerido en la Figura 123. Al hacer click en transferir, se redirige al balance de gastos. Sin embargo, este no ha cambiado, ya que la transferencia debe ser confirmada antes de considerarse como válida. Esto se debe a que las transferencias son instancias especiales de gastos, y por lo tanto deben seguir el mismo proceso de confirmación antes de ser considerados como válidos y *pagados*. Las transferencias pueden verse en el listado de gastos por confirmar:

Gastos

Gráficos .xls

Nuevo Gasto Pendientes **Por Confirmar** Pagados

Mostrar 10 registros Buscar:

Fecha	Categoría	Monto
16 de Junio de 2016	Transferencia	\$ -617
16 de Junio de 2016	Transferencia	\$ 617

Figura 127. Transferencias son visibles en el listado de gastos por confirmar.

Notar que se generaron 2 gastos: uno con un monto positivo y otro con un monto negativo. El detalle de estos mostrará que el gasto positivo fue pagado por *Allie* (Figura 128), mientras que el negativo aparece pagado por *Levni* (Figura 129). Recordar que fue *Allie* quien realizó la transferencia, por lo que hace sentido que se esté reduciendo el monto total de *Levni*, y aumentando el de *Allie*.

Monto:	\$ 617
Creado por:	Allie
Usuario:	Allie

Figura 128. Gasto generado automáticamente para *Allie*.

Monto:	\$ -617
Creado por:	Levni
Usuario:	Levni

Figura 129. Gasto generado automáticamente para *Levni*.

Naturalmente, el gasto de *Levni* debe ser confirmado por *Allie*, mientras que el de *Allie* debe ser confirmado por *Levni*.

Es necesario que los gastos sigan el mismo proceso de validación y confirmación que los gastos debido que el usuario al que se le transfiera debe confirmar que efectivamente se le pagó lo que se le debía. De otra forma, un usuario moroso puede inventar transferencias con tal de mantener el balance en 0, siendo que no realizó realmente las transferencias.

Una vez que ambas transferencias son confirmadas por los usuarios respectivos, se puede ver que el balance de gastos ha cambiado, y se encuentra equilibrado (Figura 130).

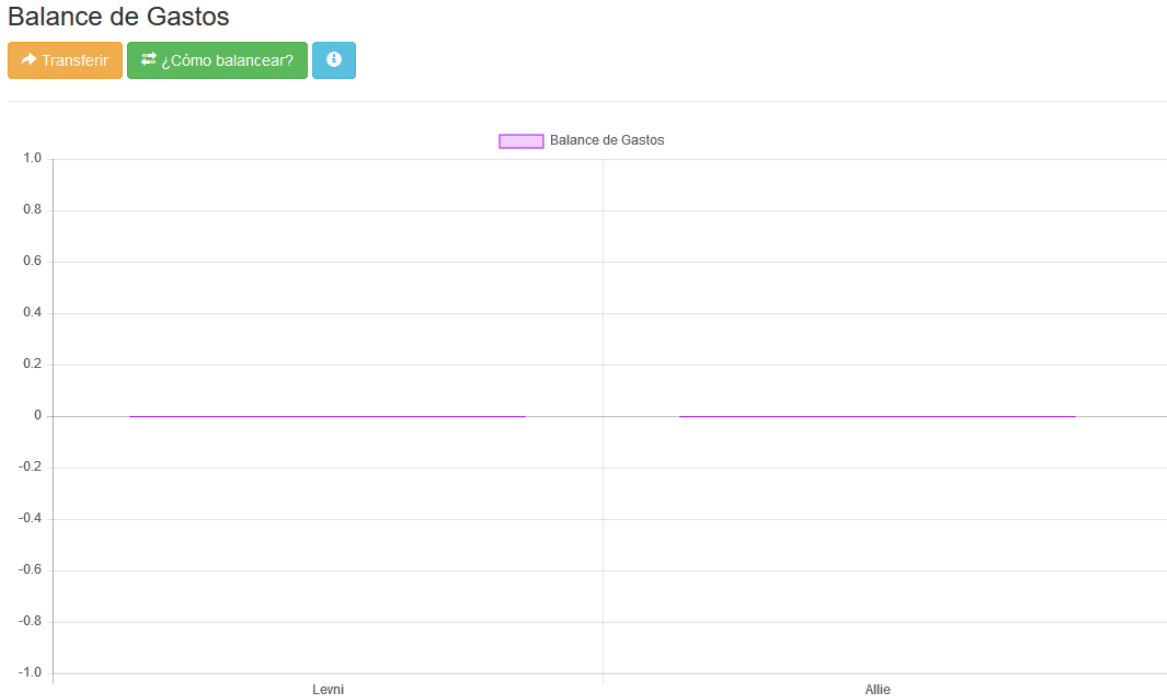


Figura 130. Una vez que se confirman las transferencias apropiadas, el balance de gastos indica que los usuarios se encuentran en equilibrio.

6. Evaluación de la solución

Para esta sección, se realizó una marcha blanca durante los meses de mayo y junio con usuarios reales. Algunos fueron invitados personalmente, mientras que otros fueron invitados por medio de un correo masivo, el cual incluía a los usuarios que dejaron sus correos durante la encuesta realizada en enero³².

De esta forma, se invitó a alrededor de 30 usuarios, de los cuales 11 registraron una cuenta, y de estos sólo 4 fueron relativamente activos durante la marcha blanca.

³² Ver Anexo A.

Las siguientes subsecciones muestran un resumen de los comentarios y sugerencias que llegaron gradualmente en forma de correos por parte de los usuarios, tanto los que usaron activamente la plataforma como aquellos que optaron por desecharla.

6.1. Sobre la interfaz y los primeros pasos

La mayor parte de los comentarios y críticas tenían relación con la interfaz y su usabilidad. Como evidencia de esto, varios usuarios fueron capaces de crear y activar sus cuentas, pero al ingresar no resultaba inmediatamente evidente que no pertenecían a una vivienda, y que debían partir por crear una. De esta forma, dichos usuarios desecharon rápidamente la idea de que la aplicación podía serles de utilidad, pues, al no tener una vivienda, no vieron la forma de ingresar gastos.

Por otro lado, aquellos usuarios que comprendieron que debían partir por crear una vivienda se encontraron con un nuevo problema: la página de configuración. Esta interfaz fue diseñada para tener acceso a todas las distintas funcionalidades, y por lo tanto contiene mucha información. Sin embargo, los usuarios nuevos no comprendían bien la nomenclatura de las distintas componentes. Vale decir, para un usuario que está recién conociendo la aplicación, la palabra *gasto* no significa necesariamente lo que se entiende por *gasto* en el contexto de este proyecto.

De esta forma, a los nuevos usuarios se les pedía inmediatamente, desde la página de configuración, que se familiarizasen con todas las componentes del proyecto (desde gastos hasta vacaciones). Es más, esto se hace incluso antes de que puedan ingresar un gasto.

6.2. Gastos y categorías

Los usuarios se mostraron agradecidos de partir con un set inicial de categorías, y más aún de poder ingresar nuevas categorías personalizadas. Sin embargo, un gran problema con este punto consiste en que un gasto sólo se puede crear utilizando una categoría pre-existente. Es decir, no es posible crear una nueva categoría directamente al momento de crear un gasto. De esta forma, si el usuario se encuentra creando un gasto y al momento de elegir la categoría cae en cuenta de que ninguna opción es la adecuada, el usuario debe:

- dirigirse a la página de creación de categorías (perdiendo los datos que ya había ingresado en el formulario de gasto)
- crear la nueva categoría
- volver a crear el gasto desde cero, pero esta vez teniendo la categoría deseada.

Este punto evidencia un problema mayor con la aplicación en general, el cual consiste en que, para que la aplicación resulte usable y no quite más tiempo del necesario, es necesario dedicarse a realizar todas las configuraciones necesarias (tanto categorías como ítems) de forma previa.

Por último, cabe mencionar que este proceso de creación de categoría no resulta tan engorroso, pues las categorías personalizadas suelen ser muy pocas, y en general no aparecen nuevas categorías con el tiempo. Sin embargo, un problema similar ocurre con los ítems, como se verá en la siguiente subsección.

6.3. Listas de compras e ítems

Las listas de compras fueron la funcionalidad más agradecida y utilizada por parte de los usuarios. Debido a esto, es justamente aquí donde se concentran la mayor parte de las críticas, pues los usuarios realmente deseaban que este módulo funcionase de la forma más expedita posible.

Todos los usuarios se mostraron contentos con la idea de tener un conjunto de ítems asociados a la vivienda, los cuales podían ser reutilizados en nuevas listas de compras, y estaban de acuerdo en que de esta forma se facilitaba ampliamente el ingreso de ítems a las listas. Sin embargo, el problema de las listas de compras recae en la usabilidad de los formularios para crear y editar la lista. Por un lado, se tiene un problema muy similar al de las categorías y gastos; sólo se pueden agregar ítems pre-existentes a la lista. Vale decir, no se permite ingresar un ítem nuevo, y que el servidor lo cree al procesar el formulario. El usuario está obligado a elegir el ítem de un listado, y si no encuentra el que desea, debe ir a la página de ítems para crear uno nuevo. Esto, al igual que en el caso de los gastos y categorías, tiene la gran desventaja de que se pierde la lista que se estaba creando, y se debe comenzar desde cero. Sin embargo, para el caso de los ítems es mucho más engorroso, pues es más común crear ítems que categorías.

Por último, existe otro grave problema con la usabilidad de la validación al crear una nueva lista; si ésta contiene errores, se crea un mensaje de error con estos, pero el usuario debe comenzar a crear la lista desde cero.

Si bien este problema se abordó brevemente en algún punto del desarrollo, se subestimo su importancia, siendo considerado un problema menor. Sin embargo, llegado el momento de la marcha blanca, rápidamente los usuarios manifestaron su molestia, pues lo consideraban un problema bastante serio de usabilidad, más aún si se toma en cuenta que las listas de compras corresponden a la parte más utilizada.

Notar que esto sólo ocurre al momento de crear la lista, y no al momento de editarla. Esto se debe a que el formulario para editar fue construido utilizando correctamente las componentes que Django provee para ese tipo de validaciones, mientras que el formulario de creación se valida utilizando métodos personalizados. Esto corresponde a un uso pobre del framework, cosa que, a su vez, dificulta escalar el proyecto.

6.4. Sobre la reutilización de presupuestos

Debido a la corta duración de la marcha blanca, fueron pocos los usuarios que realmente hicieron uso de los presupuestos. Sin embargo, aquellos que lo hicieron se mostraron bastante conformes

con la representación por medio de gráficos de torta y con la creación de presupuestos en general. Dichas interfaces resultaban fáciles de utilizar tanto desde sus dispositivos móviles como del navegador de escritorio.

Por otro lado, la interfaz para reutilizar los presupuestos fue fuertemente criticada, siendo incluso cuestionada su utilidad, al menos con la implementación que existe actualmente. En primer lugar, al momento de reutilizar un presupuesto sólo se carga el monto del presupuesto, pero no se muestra el total real invertido en ese período. Es decir, no se muestra qué tan acertado fue el presupuesto para el período indicado. Esta información resulta fundamental al momento de decidir si se desea o no reutilizar un presupuesto anterior.

Además, al momento de pasar los presupuestos de un período al siguiente, es necesario crearlos uno por uno, en lugar de poder seleccionar varios y copiarlos directamente al nuevo período. Es más, los períodos futuros no existen en la base de datos hasta el momento en que se produce el cambio de mes, por lo que necesariamente todos los períodos se inicializan sin ningún presupuesto definido. Luego, debería existir, por lo menos, un botón para copiar todos los presupuestos del período anterior.

6.5. Sobre el balance y las transferencias

Al igual que ocurrió con los presupuestos, ningún usuario consideró particularmente útiles los gráficos de gastos, pues sólo se podían graficar 1 o 2 períodos (mayo-junio). De la misma forma, ningún usuario llegó a utilizar el cálculo de balances, ya que les resultó difícil convencer a todos sus compañeros que utilicen la aplicación, y en consecuencia eran los únicos usando la plataforma en su vivienda. De esta forma, no se comprobó la utilidad de la implementación actual de esta función.

A pesar de esto, hubo muchos comentarios acerca de lo útil que podría llegar a ser esta funcionalidad, sugiriendo incluso extraer el cálculo de balances y transferencias e implementarlo como una función “rápida”, donde se pueda ingresar los gastos de los usuarios, sin tener que estar todos los usuarios registrados.

Es decir, suponiendo que se tiene un usuario usando la aplicación y dos sin usarla, y suponiendo que se tienen calculados los totales que cada uno ha invertido (usando los métodos externos que ya se usaban en su vivienda), se propone que el usuario registrado sea capaz de ingresar dichos totales (marcando a los otros usuarios con un alias o un usuario temporal), para luego generar las instrucciones deseadas.

6.6. Otros comentarios

Muchos usuarios manifestaron que el proyecto funcionaría mejor si fuese una aplicación móvil, en lugar de una aplicación web. Este punto mostró ser un factor determinante al momento de

fomentar el uso de la aplicación, ya que los usuarios están mucho más dispuestos a ingresar a una *app* que a una página web en el navegador del smartphone.

Al mismo tiempo, varios usuarios mostraron dudas acerca de la seguridad del ingreso de datos, a pesar de que se les aseguró que los demás usuarios no podrían ver sus gastos ni fotos de boletas. Esto va de la mano con el punto anterior, ya que en general los usuarios confían en una *app* que está en la tienda oficial de aplicaciones³³, cosa que no ocurre con un dominio web que les resulta extraño. Era de esperar que los usuarios se preocuparan respecto a esto, ya que los datos son de carácter sumamente sensible, y si bien se resolvieron estos problemas en el código, al parecer no se tuvo éxito al momento de convencer a los usuarios de que dicha seguridad existía.

Como comentario final, algunos campos en los formularios no resultan fáciles de usar. Por ejemplo, no resulta fácil ingresar el período en un nuevo gasto, ya que se muestra el listado de períodos en lugar de un *datepicker*, cosa que tendría mucho más sentido dado que se trata de una fecha.

7. Conclusiones y trabajo futuro

En este trabajo de memoria se identificó un problema que afecta actualmente a muchos estudiantes que estudian fuera de su región de origen, el cual consiste en el manejo compartido de gastos originados por una vivienda. Se propuso una solución tecnológica a este problema, para luego realizar una implementación de dicha solución. Por último, se llevó a cabo una marcha blanca durante los meses de mayo y junio de 2016 con usuarios reales.

Si bien no fueron muchos los usuarios que optaron por darle una oportunidad a la plataforma en su versión actual, se pudo comprobar que los usuarios estaban interesados en una solución tecnológica al problema planteado, pero la implementación no resultó lo suficientemente amigable como para funcionar en un entorno de uso real.

Sin embargo, en el tiempo que duró la marcha blanca fue posible recolectar mucho feedback por parte de los usuarios, el cual apunta principalmente al tema de la usabilidad y del ingreso de datos por parte de los mismos usuarios, más que a un replanteamiento completo del problema o de la solución.

Cabe mencionar que se intentó adelantar muchas funcionalidades que podrían resultar atractivas para los usuarios, lo que llevó a una página de configuración muy compleja, y a un período de desarrollo más largo. Esto retrasó, a su vez, el comienzo de la marcha blanca. Una estrategia alternativa para abordar este desarrollo podría haber sido partir con el ingreso de datos (gastos, categorías, listas e ítems), y luego lanzar la marcha blanca uno o dos meses antes para ir descubriendo así las funcionalidades que realmente fuesen a usar los usuarios.

³³ Vale decir, en la *AppStore* (dispositivos iOS) o en la *PlayStore* (dispositivos Android).

Por último, se comprobó que una solución tecnológica al problema planteado es factible y útil, siempre y cuando sea suficientemente *user-friendly*, y cumpla con los estándares de usabilidad a la que los usuarios actuales de aplicaciones móviles se han acostumbrado. Sumado a esto, se considera que, si bien la implementación descrita en este documento no logró resolver en toda su magnitud el problema planteado, los servicios implementados son útiles y sirven como una base sólida sobre la cual construir una solución más usable.

La implementación actual de la plataforma no consiste de ninguna forma en el producto final, ya que se pretende continuar con el desarrollo del proyecto, el cual se adecuará a las necesidades de los usuarios. De esta forma, hay mucho espacio para el crecimiento del proyecto, partiendo por el feedback discutido en la evaluación de la solución.

Por un lado, se pretende simplificar ampliamente la página de configuración, haciendo una división entre las funciones *básicas* (ingreso de gastos, categorías, ítems y listas), y las funciones *avanzadas* (balances, transferencias y vacaciones). De esta forma, se busca que los usuarios se sientan más cómodos durante las primeras semanas de uso. Por otro lado, resulta necesario permitir el ingreso de nuevos ítems en el formulario de listas, y de nuevas categorías en el de gastos.

Finalmente, se pretende implementar, además, una aplicación móvil, inicialmente en *Android*. Esto resulta necesario ya que facilitaría ampliamente el uso de la aplicación desde un smartphone, y al mismo tiempo daría seguridad a los usuarios respecto a la confidencialidad de los datos ingresados, ya que la aplicación estaría disponible en la *PlayStore*. Este punto conlleva además un desafío adicional, ya que se pretende tener un mismo servidor para la versión web y la versión móvil, por lo que resulta lógica la implementación de una API REST que facilite el desarrollo de la versión *Android*, así como una posible versión en *iOS*, en caso de ser necesaria en el futuro.

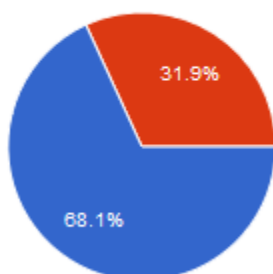
8. Bibliografía

- [1] Facultad de Ciencias Física y Matemáticas, Universidad de Chile. Procedencia de alumnos. URL: <http://ingenieria.uchile.cl/admision/admision-regular-pregrado/91180/procedencia-de-alumnos>. Última revisión: Junio 2016.
- [2] Percival, H., 2014. *Test-Driven Development with Python*. O'Reilly Media, Inc.
- [3] Greenfeld, D. & Greenfeld, A., 2015. *Two scoops of Django: best practices for Django 1.8*. Corona, CA: Two Scoops Press.

Anexo A. Resultados de la encuesta web

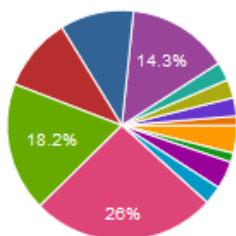
A continuación, se muestran los resultados de la encuesta web realizada, la cual contó con un total de 113 respuestas al 13 de enero de 2016:

Vives en una vivienda compartida?



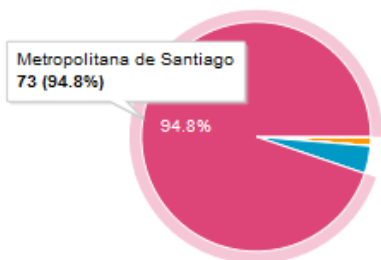
Sí	77	68.1%
No	36	31.9%

¿De dónde eres?



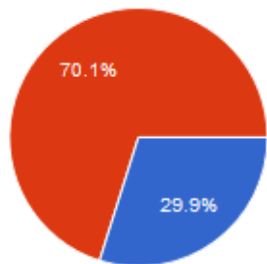
Arica y Parinacota	0	0%
Tarapacá	0	0%
Antofagasta	3	3.9%
Atacama	1	1.3%
Coquimbo	3	3.9%
Valparaíso	2	2.6%
Metropolitana de Santiago	20	26%
O'Higgins	14	18.2%
Maule	8	10.4%
Bío-Bío	8	10.4%
La Araucanía	11	14.3%
Los Ríos	2	2.6%
Los Lagos	2	2.6%
Aysén	2	2.6%
Magallanes	1	1.3%

¿Dónde estudias?



Arica y Parinacota	0	0%
Tarapacá	0	0%
Antofagasta	1	1.3%
Atacama	0	0%
Coquimbo	0	0%
Valparaíso	3	3.9%
Metropolitana de Santiago	73	94.8%
O'Higgins	0	0%
Maule	0	0%
Bío-Bío	0	0%
La Araucanía	0	0%
Los Ríos	0	0%
Los Lagos	0	0%
Aysén	0	0%
Magallanes	0	0%

En la vivienda, ¿Tienen algún sistema para administrar los gastos de cada uno?



Sí	23	29.9%
No	54	70.1%

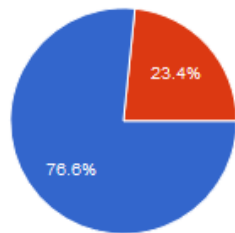
Si la respuesta anterior fue sí, ¿Podrías describir brevemente el sistema?

- Cada una es responsable de sus propios gastos en cuanto a comida y útiles de aseo, lo único que es comunitario son cosas puntuales como el confort, los fósforos y los productos de limpieza. Cuando hay que comprar alguno de estos, lo compra la primera que pueda, después divide en 3 el precio (somos 3 en el dpto.) y las otras 2 le reembolsan su parte. Usamos un sistema parecido para las cuentas, yo pago todo de mi bolsillo a medida que llegan y ellas me reembolsan a principios del otro mes
- Sólo las cuentas, en un Excel.

- En un cuaderno anotamos mensualmente quiénes han pagado la cuota acordada (\$2.500) y qué cosas se compran con ese dinero, adjuntando boleta y anotando los detalles.
- Una persona compra, se divide el valor de la compra en el total de inmediato.
- Una planilla Excel.
- Pegamos las boletas en el refrigerador y luego usamos Tricount para dividir los gastos, considerando quién usó qué cosas de lo que compramos; por ejemplo, yo no como queso, así que eso yo no lo pago.
- Llevamos las cuentas en una libreta (luz agua gastos comunes supermercado... etc.).
- Planilla Excel básica.
- Tenemos un fondo común, y cada uno se acuerda más o menos lo que gasta. Además, para las cuentas hay una planilla de Excel.
- Hay una responsable, ella nos avisa de las cuentas y los gastos y divide.
- Se dividen las cuentas entre dos.
- Excel.
- Se suman los gastos, y se dividen por 3, comida individual.
- Cuentas (gastos comunes, agua, luz, gas e internet) se dividen en partes iguales entre las que compartimos el departamento, cada una compra lo que consume (proteínas, pan, jamón, queso, yogur, cereales, etc.) mientras que las cosas de despensa más generales (azúcar, sal, aceite, arroz, aliños, té, sopas, jugos, etc.), aseo e higiene (lavalozas, servilletas, confort, pasta de dientes, etc.), las compramos alternadamente, pero sin llevar registro de eso, solo a conciencia de cada una (llevo dos años en el mismo sistema y ha funcionado bien). Normalmente cada una compra su fruta y verdura, pero en ocasiones nos envían nuestras familias en mayor cantidad y se comparte, cuando es así, nos avisamos.
- Nos compran un pedido de supermercado al mes y tenemos un fondo común para comprar cosas para cuando faltan, además nos turnamos los días para comprar pan.
- Hay una encargada de gastos, mensualmente nos envía las fotos de las cuentas por WhatsApp y pagamos. Sin embargo, no tenemos ningún tipo de respaldo ni de registro (por ej. planilla Excel).
- Excel.
- Me pagan mensualmente y yo pago todo.

- Google doc
- A medias.
- Excel con los gastos, calcula lo que debe aportar cada uno.
- Un Google sheet.

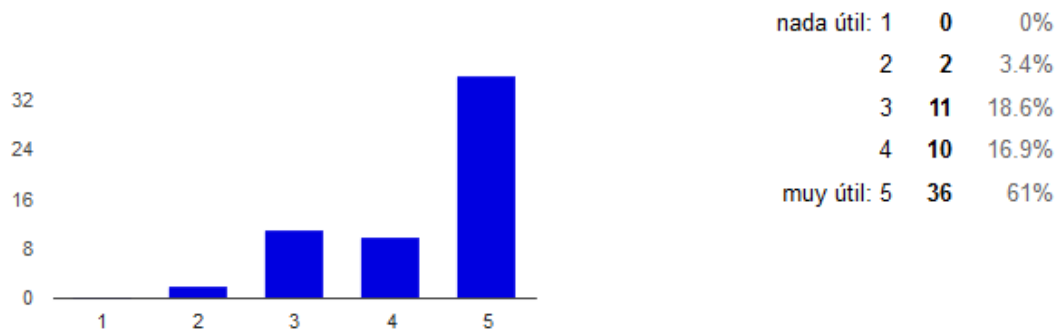
¿Crees que sería beneficioso tener un sistema web para administrar los gastos compartidos?



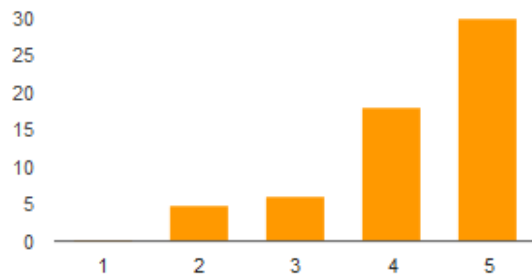
Sí	59	76.6%
No	18	23.4%

¿Qué tan útiles crees que serían las siguientes funcionalidades?

Balanceo de los gastos de cada integrante

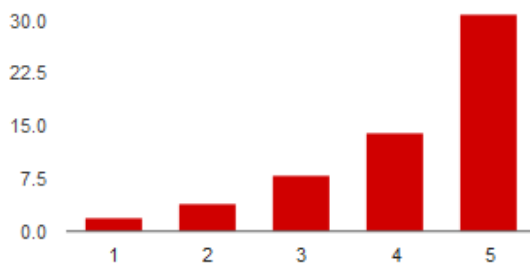


Gráficos de gastos por categoría por período



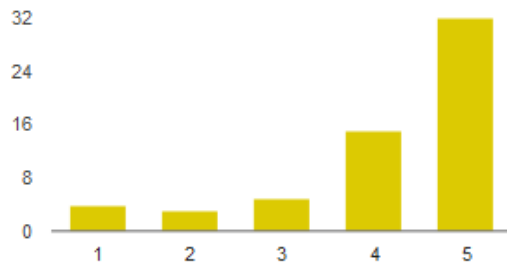
nada útil: 1	0	0%
2	5	8.5%
3	6	10.2%
4	18	30.5%
muy útil: 5	30	50.8%

Gráficos de gastos por integrante por período



nada útil: 1	2	3.4%
2	4	6.8%
3	8	13.6%
4	14	23.7%
muy útil: 5	31	52.5%

Lista de compras online



nada útil: 1	4	6.8%
2	3	5.1%
3	5	8.5%
4	15	25.4%
muy útil: 5	32	54.2%

Por último, un total de 24 personas se mostraron dispuestas a participar de las primeras versiones en calidad de *beta-tester*, por lo que incluyeron sus e-mails en las respuestas.