



Refactoring Centrado en el Usuario

Dra. Alejandra Garrido

garrido@lifa.info.unlp.edu.ar

CONICET / LIFIA, Fac. de Informática

Univ. Nac. de La Plata, Argentina



Presentación realizada por [Alejandra Garrido](#) bajo licencia [Creative Commons Atribución-NoComercial 4.0 Internacional](#)

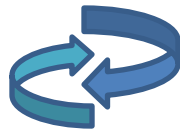
Contenido

- Algo sobre mi
- Refactoring para la mejora incremental de la calidad interna
- Proceso de refactoring: bad smells
- Desafíos existentes en la mejora incremental de la UX
- Refactoring para la mejora incremental de la UX
- UX smells y UX refactorings
- Herramientas
- Conclusiones y desafíos por delante

Sobre mi...



- Me uní al LIFIA, Fac. de Informática, UNLP, Argentina en 1994
- Comencé mi investigación y realicé mi tesis de grado en relación a frameworks OO y patrones de diseño en el área de hipermedia
- Realicé mi MS y PhD en la Univ. Illinois at UC en code refactoring
- Me reinserté en el LIFIA en 2006 trabajando en:
 - Refactoring
 - Web application design
- Proyecto actual que dirijo: [UX-Driven Development](#) –
Monitorización continua de la experiencia del usuario durante el desarrollo y mantenimiento ágil de aplicaciones interactivas



Refactoring interno

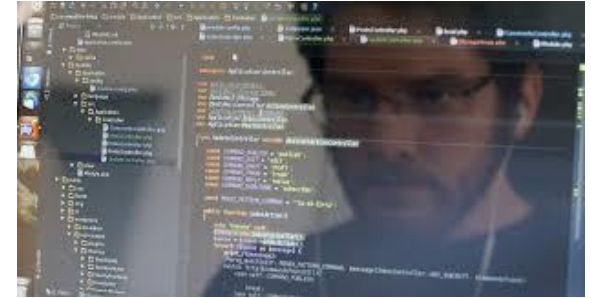


- Aparece como respuesta a la necesidad de adaptarse a los cambios en un proyecto
- Técnica que permite cambiar la estructura interna del software para mejorar:
 - **calidad interna:** legibilidad, adaptabilidad, mantenibilidad, reusabilidad
 - sin cambiar el comportamiento externo (Fowler, “*Refactoring*”. Addison-Wesley. 1999)



Proceso de Refactoring

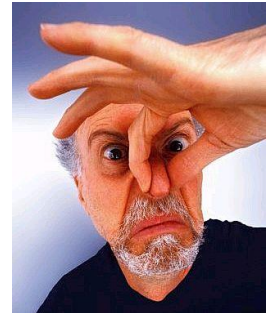
- Implica
 - Simplificar lógicas complejas
 - Eliminar duplicaciones
 - Reorganizar responsabilidades entre componentes
- A través de cambios *pequeños*
 - Ejemplos: Extract Method, Rename, Move Method, etc.
- Refactoring para y por los programadores



Dónde aplicar refactoring?

- **Malos olores (*code smells*):**

Estructuras en el código que sugieren la posibilidad / necesidad de un refactoring



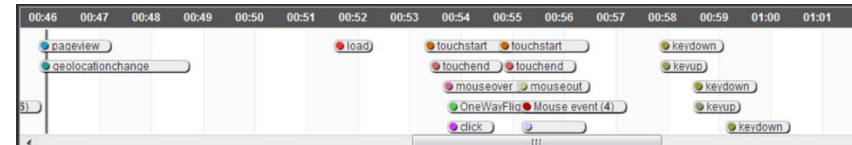
- Ejemplos:

- *Duplicated code* ➞ Extract Method; Pull Up Method
- *Large class* ➞ Extract Subclass
- *Método largo* ➞ Extract Method
- *Sentencias condicionales* ➞ Decompose Conditional

Cómo lograr la mejora continua de UX



Centrarse en el usuario



?

¿Y si encuentra problemas?

Desafíos de la mejora incremental de UX

- Los diseñadores de UX necesitan poder trabajar a la par que los desarrolladores, sincronizando sus prácticas
- Los diseñadores UX necesitan poder ***innovar***, probar soluciones, probar nuevas ideas
- Queremos poder:
 - ***aprender del feedback***
 - ***fallar rápido***
 - ***adaptarnos más rápido***

A/B testing y refactoring de UX



(“Usability Improvement through A/B testing and refactoring”. Firmenich, Garrido, Grigera, Rivero, Rossi. Software Quality Journal 27(1). 2019)

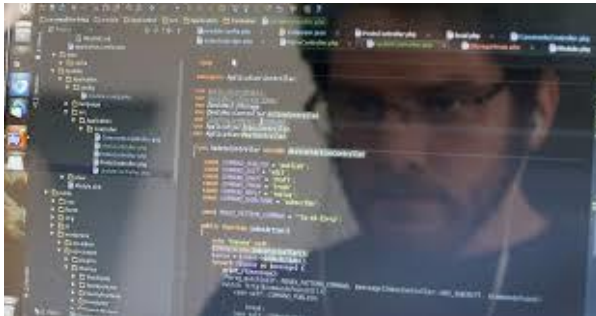
☞ Refactorings para mejorar calidad externa

- Propósito: mejorar
 - usabilidad
 - accesibilidad
 - UX
- Alcance:
 - navegación
 - presentación
 - interacción del usuario
- Sin modificar la funcionalidad de la aplicación



*(“Refactoring For Usability In Web Applications”.
Garrido, Rossi, Distant. IEEE Software. May/June 2011)*

Refactoring de código



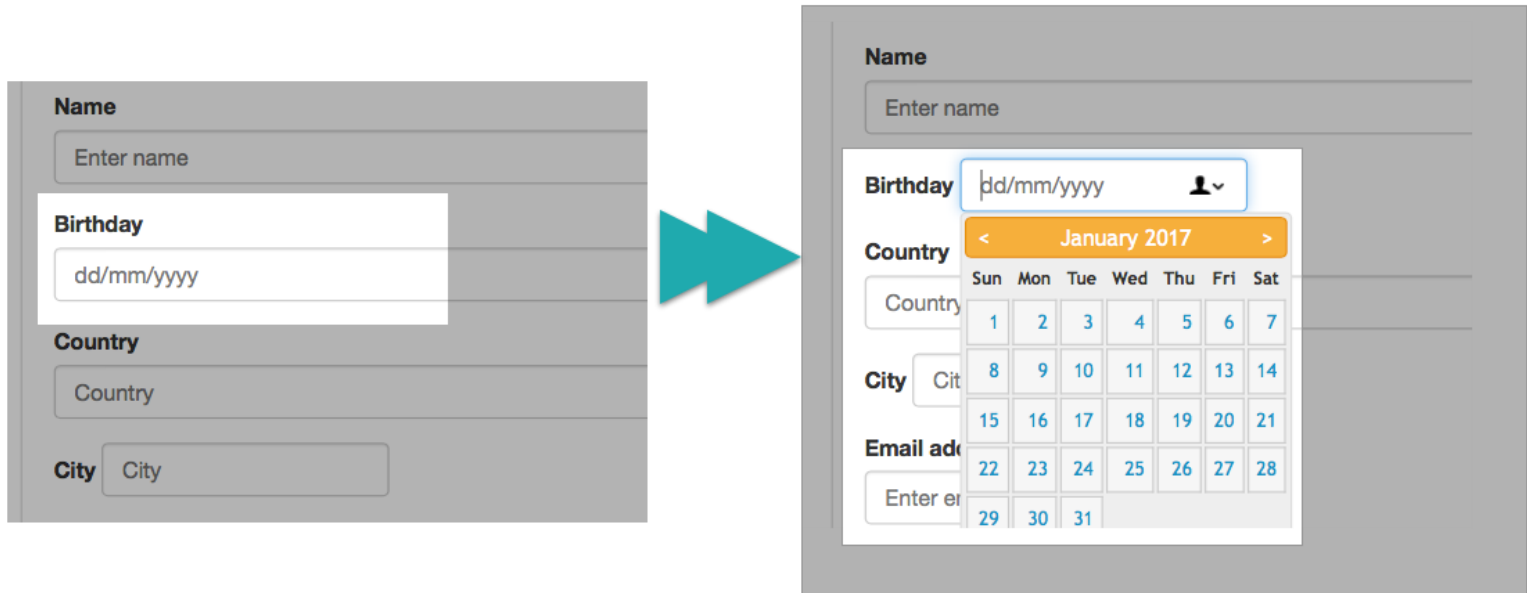
- Calidad interna
- Para los desarrolladores

Refactoring de IU



- Calidad externa
- Para los usuarios

Refactoring: Add Date Picker



The diagram illustrates a refactoring process for a user form. On the left, the initial form has fields for Name, Birthday (with a placeholder 'dd/mm/yyyy'), Country, and City. A large teal arrow points to the right, where the refactored form is shown. In the refactored version, the Birthday field is replaced by a date picker widget. This widget includes a text input with the placeholder 'dd/mm/yyyy', a user icon, and a calendar dropdown for January 2017. The calendar shows a grid of days from 1 to 31. The other fields (Name, Country, City, Email address) remain unchanged in their layout and labels.


Name
Enter name


Birthday
dd/mm/yyyy

Country
Country

City City

Email address
Enter email

Birthday dd/mm/yyyy 

Country 

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				


Los “malos olores” de las IU


Unresponsive Element

Información actual del pedido

Las órdenes enviadas dentro o fuera de Argentina se facturan en **Pesos**. Para obtener un valor estimado en **Dólares**, la cotización actual es (\$ 3,92 Peso Argentino = US\$ 1.00 Dolar)

Borrar	Título	Cantidad	Precio
<input type="checkbox"/>	INGENIERIA DE SOFTWARE. Normalmente salida del depósito en 3 días	<input type="text" value="1"/>	\$ 112,00
<input type="checkbox"/>	INGENIERIA DEL SOFTWARE. En Stock. Salida del depósito en 48 horas	<input type="text" value="1"/>	\$ 154,00
<input type="checkbox"/>	PATRONES DE DISEÑO. En Stock. Salida del depósito en 48 horas	<input type="text" value="1"/>	\$ 139,00
	El peso de su orden es 2,99 Kg	Subtotal	\$ 405,00

 [Información acerca de Gastos de Envío y Tiempo de Entrega](#)

 [Ir a la caja](#) Presione este botón para informar la dirección de envío y medio de pago.

Refactoring: Turn Attribute into Link


(to solve Unresponsive Element)

Información actual del pedido

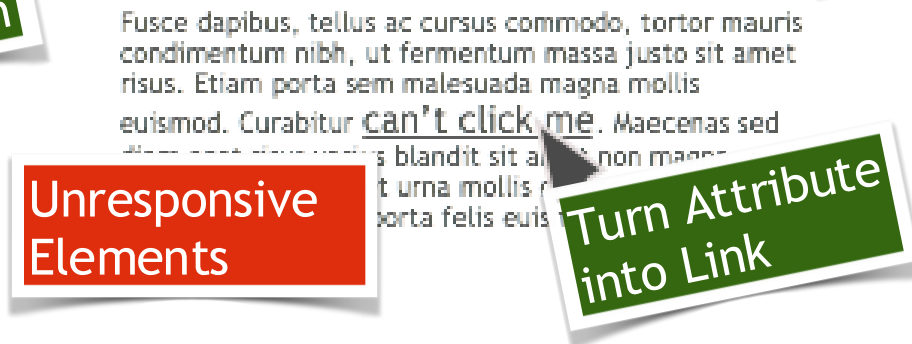
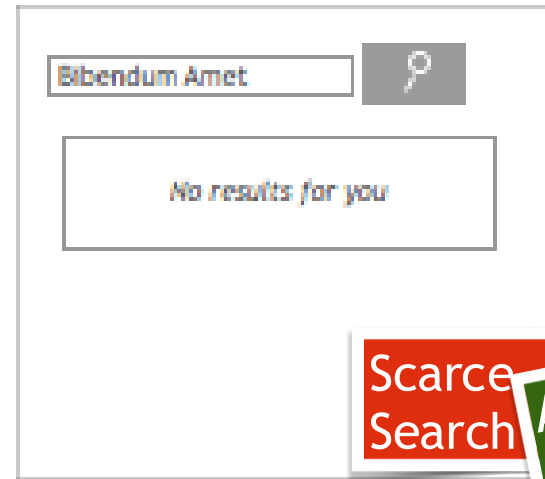
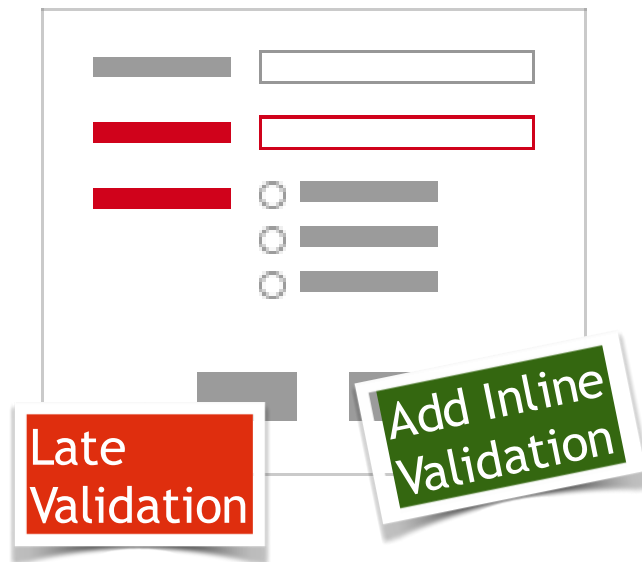
Las órdenes enviadas dentro o fuera de Argentina se facturan en **Pesos**. Para obtener un valor estimado en **Dólares**, la cotización actual es (\$ 3,92 Peso Argentino = US\$ 1.00 Dolar)

Borrar	Título	Cantidad	Precio
<input type="checkbox"/>	INGENIERIA DE SOFTWARE. Normalmente salida del depósito en 3 días	<input type="text" value="1"/>	\$ 112,00
<input type="checkbox"/>	INGENIERIA DE SOFTWARE. En Stock. Salida del depósito en 48 horas	<input type="text" value="1"/>	\$ 154,00
<input type="checkbox"/>	PATRONES DE DISEÑO. En Stock. Salida del depósito en 48 horas	<input type="text" value="1"/>	\$ 139,00
	El peso de su orden es 2,99 Kg	Subtotal	\$ 405,00

 [Información acerca de Gastos de Envío y Tiempo de Entrega](#)

 [Ir a la caja](#) Presione este botón para informar la dirección de envío y medio de pago.

UX Refactorings



Refactoring para mejorar la accesibilidad universal

Ejemplos de accessibility refactorings:

- *Replace non-accessible menu by list of links*
- *Eliminate Duplicated Content*
- *Add Size Indicators*
- *Split page*
- *Distribute Menu*
- *Postpone Selection*

(“Personalized Web Accessibility using Client-Side Refactoring”.
Garrido et al. IEEE Internet Comput. 17(4): 2013)

(“Improving accessibility of Web interfaces: refactoring to the rescue.”.
Garrido et al. UAIS 13(4) 2014)



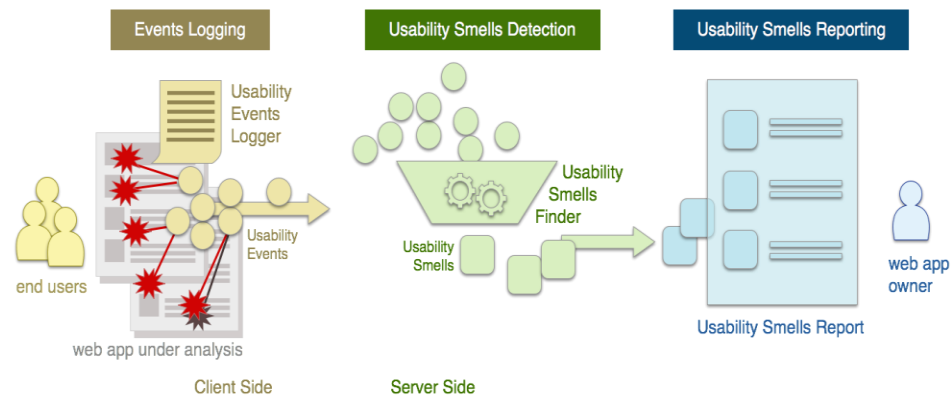
Automatización: Usability Smell Finder (USF) - Kobold

- Detecta malos olores en la interacción del usuario con la aplicación web

*(“Automatic detection of usability smells in web applications”,
Grigera, Garrido, Rivero, Rossi. IJHCS 97, 2017)*

- Sugiere soluciones a través de refactorings

*(“Kobold: Web Usability as a Service”,
Grigera, Garrido, Rossi. ASE Demo. 2017)*



Una herramienta para el diseñador: UX-Painter

Permite que el diseñador UX pueda crear distintas versiones de una aplicación web a través de programación visual, aplicando CSWR (Client-Side Web Refactorings)

The image displays three distinct versions of a 'Personal Information' form, illustrating the flexibility of UX-Painter in creating multiple UI designs from a single conceptual model.

- Top Left Form:** A simple form with three stacked text input fields labeled 'Name', 'Birthday', and 'Gender'.
- Bottom Left Form:** A form with a 'Name' text input, a 'Birthday' text input with a date mask (____-__-__), and 'Gender' selected via radio buttons for 'Female', 'Male', and 'Other' (which has an associated text input).
- Right Form:** A form with a 'Name' text input, a 'Birthday' section with three separate dropdown menus for 'Day', 'Month', and 'Year', and a 'Gender' dropdown menu with options 'Male', 'Female', and 'Other'. A 'Phone' label is also visible below the gender dropdown.

(“UX-Painter: An Approach to Explore Interaction Fixes in the Browser”, Gardey, Garrido, Firmenich, Grigera, Rossi. Proc. ACM Hum. Comput. Interact. 4(EICS): 89:1-89:21 - 2020)

Conclusiones

“Refactoring: Our only defense against software decay”
(Ralph Johnson)

- Refactoring es esencial para mantener calidad del software, tanto interna como externa
- El refactoring de UX del lado del cliente nos permite:
 - aprender del feedback del usuario
 - agregar patrones luego de que el feedback demuestra que son necesarios
 - adaptarnos a los cambios en forma rápida y segura

Conclusiones – Desafíos por delante

- Incrementar el catálogo de UX smells y UX refactorings para aplicaciones web y móviles
- Definir nuevas métricas para comparar refactorings alternativos automáticamente

*(“One Metric for All: Calculating interaction effort of individual widgets”
Grigera, Gardey, Rodriguez, Garrido, Rossi. CHI 2019 Extended Abstracts)*

- Fomentar la adopción de procesos de mejora de la calidad centrada en el usuario

*(UX-Driven Development – Monitorización continua de la UX
durante el desarrollo y mantenimiento ágil de aplicaciones interactivas. PICT 2021)*



alejandra.garrido@lifa.info.unlp.edu.ar

<https://lifa.info.unlp.edu.ar/dra-alejandra-garrido/>

<https://www.linkedin.com/in/alejandragarrido/>