



TESINA DE LICENCIATURA

Título: Aumentación de sitios web combinando enfoques MDWE y técnicas de separación de concerns.

Autores: Pedro Maglione Sarasibar

Director: Mario Matías Urbietta

Codirector: Sergio Firmenich

Asesor profesional:

Carrera: Licenciatura en sistemas

Resumen

La aumentación web es un conjunto de técnicas que permiten a los usuarios definir y ejecutar software que depende de la capa de presentación de una página web concreta. De esta manera, a través del uso de artefactos de aumentación web, los usuarios finales pueden satisfacer varios tipos de requisitos que no fueron considerados por los analistas, desarrolladores e interesados que construyeron la aplicación. Aunque hay algunos enfoques de aumento que contemplan una contraparte en el servidor (para soportar aspectos tales como colaboración, gestión de sesión de explorador cruzado, etc.), los artefactos de aumentación suelen ser puramente del lado del cliente. Este soporte del lado del servidor aumenta las capacidades de las ampliaciones, ya que puede permitir compartir información entre usuarios y dispositivos.

Este trabajo presenta un nuevo enfoque para el diseño de aplicaciones de aumentaciones web basado en el lado del cliente y componentes del lado del servidor. Se propone un enfoque basado en el modelo que eleva el nivel de abstracción para el desarrollo del servidor. El enfoque utiliza principios avanzados de separación de conceptos, por lo que se proporcionan un conjunto de herramientas para diseñar la composición de la aplicación del núcleo y el aumento. Las ideas y enfoque se ilustran con varios ejemplos corrientes que muestran el potencial del enfoque.

Palabras Claves

Aumentación web, interfaz de usuario, DOM, cliente-servidor, Model-Driven Web Engineering, Web Ratio, Web Object Ambient, IFML: The Interaction Flow Modeling Language

Trabajos Realizados

Con el fin de evaluar la viabilidad de este enfoque, se llevaron a cabo un estudio de usabilidad de las aplicaciones aumentadas con los usuarios finales. El objetivo es analizar los aumentos de la web que se basan en las características del servidor para medir cómo se mejora la usabilidad en contraste con la misma tarea realizada sin el aumento. El estudio se llevó a cabo desde el punto de vista de los investigadores en el contexto de los usuarios finales que diariamente usan aplicaciones web.

Conclusiones

En este trabajo, se presenta y prueba mediante ejemplos un enfoque novedoso para diseñar aumentaciones web con comportamientos del lado del cliente y del servidor. El enfoque se basa en los principios de la separación de conceptos, por lo tanto, se proporciona el mecanismo de composición para cada modelo (Conceptual, navegacional y de interfaz de usuario). Los resultados de este trabajo presentan una evidencia preliminar de que un aumento con soporte de servidor introduce una mejora estadísticamente significativa.

Trabajos Futuros

El experimento se basó en métricas objetivas evaluadas con todos los datos recopilados sin exclusión para garantizar resultados idempotentes y evitar la pesca de hipótesis. El bajo poder estadístico es una amenaza para este estudio. Por lo tanto, se debe realizar un estudio más amplio con un conjunto más amplio de muestras que ayuden a diluir las irrelevancias del experimento. Por otro lado, también se proponen varias mejoras en la herramienta en base a la experiencia de los usuarios.

Índice

Índice de figuras

Resumen	1
1 Introducción	2
2 Marco teórico y estado del arte	6
2.1 Aumentación de aplicaciones web	6
2.2 Ingeniería de enfoques orientados a modelos	7
2.2.1 Diseño conceptual	7
2.2.2 Diseño navegacional	7
2.2.3 Diseño de interface abstracta	8
2.3 Trabajos relacionados	8
2.3.1 Aumentación de aplicaciones web	8
2.3.2 Separación de conceptos en MDWE	9
2.4 Enfoque de aumento web impulsado por modelos	10
2.4.1 Justificación sobre el uso del enfoque	10
3 El enfoque propuesto	13
3.1 Descripción del enfoque	13
3.1.1 Extraer el modelo conceptual de la aplicación principal (Paso 2)	15
3.2 Modelado de la capa de aumentación	18
3.2.1 Modelo Conceptual (Paso 3a)	19
3.2.2 Diseño Navegacional (Paso 3b)	19
3.2.3 Enlazado Estructural de la Aumentación (Paso 3d)	21
3.2.4 Implementación en la nube	22
3.2.4.1 Herramienta de Catálogo	24

3.2.5	Aplicación de plantillas de extracción: soporte del lado del servidor y del cliente	25
3.2.5.1	Hacia una navegación con estado	25
3.2.5.2	API de Recuperación de instancias	26
3.3	Un ejemplo comprensivo	28
3.3.1	Transmisión de contenido desde el servidor	29
3.3.1.1	Extraer el modelo conceptual de la aplicación principal	29
3.3.2	Modelando la Capa de Aumentación	30
3.3.3	Orquestación compleja de conceptos sensibles a la navegación .	33
3.3.3.1	Extraer el modelo conceptual de la aplicación principal	34
3.3.3.2	Modelado de la capa de aumento	34
3.3.4	Identificación única de los usuarios por instancia	37
3.3.4.1	Identificación de usuario no persistente en el tiempo .	40
3.3.5	Manejo de excepciones en la herramienta	41
3.3.6	Mejora de las aplicaciones con información entre sitios	43
3.3.7	Implementación	45
3.3.8	Cómo se realiza el intercambio de datos entre el frontend y el backend	47
3.4	Evaluando el enfoque	48
3.4.1	Impacto	49
4	Discusión técnica	54
4.1	Introducción	54
4.2	Frontend	55
4.2.1	Requerimientos WOA	55
4.2.2	Versión de Mozilla Firefox	56
4.2.3	Peticiones desde el navegador web	57
4.3	Backend	58
4.3.1	IRAPI	58
5	Trabajos realizados	61
5.1	Objetivo	61
5.2	Sujetos	62
5.3	Materiales	62
5.4	Protocolo	63
5.5	Resultados, análisis e implicancia	65

Índice

5.6 Amenazas a la validez	65
6 Trabajos futuros	68
6.1 Experimento	68
6.2 Mejoras en la herramienta	68
7 Conclusión	70
Referencias	72

Índice de figuras

3.1	Esquema del enfoque	16
3.2	Definición del concepto	17
3.3	Esquema de implementación en la nube	23
3.4	Una lista de los aumentos disponibles en el catálogo	25
3.5	Un aumento almacenado en el catálogo	25
3.6	Capa de aumento basada en instancias extraídas de sitios web visitados	26
3.7	Capa de aumento basada en instancias extraídas por IRAPI	27
3.8	Identificación de la instancia y concepto en contenido web existente . .	29
3.9	Modelo Entidad-Relación	30
3.10	El servicio representado en Web IFML	31
3.11	Resultado de la aumentación	32
3.12	Flujo de interacción entre componentes de la aumentación	36
3.13	Modelo conceptual y navegacional de la Planificación de Producción . .	38
3.14	Navegación y resultado en la Planificación de Producción	39
3.15	Modelo conceptual de seguridad	41
3.16	Página de error n° 1	43
3.17	Página de error n° 2	43
3.18	Página de error n° 3	43
3.19	Página de error n° 4	43
3.20	Página de error del servicio de planificación de producción	44
3.21	Transcluyendo los precios del mercado central con el contenido entre sitios	46
3.22	Ejemplo de solicitud utilizando el servicio CORS	49
5.1	Edad de los sujetos	62
5.2	Puntaje SUS	66
5.3	Tiempo dedicado en realizar la tarea	67

Resumen

La aumentación web es un conjunto de técnicas que permiten a los usuarios definir y ejecutar software que depende de la capa de presentación de una página web concreta. De esta manera, a través del uso de artefactos de aumento web, los usuarios finales pueden satisfacer varios tipos de requisitos que no fueron considerados por los analistas, desarrolladores e interesados que construyeron la aplicación. Aunque hay algunos enfoques de aumentación que contemplan una contraparte de servidor (para soportar aspectos tales como colaboración, gestión de sesión de explorador cruzado, etc.), los artefactos de aumentación suelen ser puramente del lado del cliente. Este soporte del lado del servidor aumenta las capacidades de las aumentaciones, ya que puede permitir compartir información entre usuarios e incluso entre las mismas aplicaciones. Hasta ahora, este apoyo se define a menudo y se desarrolla de una manera ad-hoc. Aunque está claro que el soporte del servidor aporta nuevas posibilidades, también es cierto que el desarrollo y despliegue de aplicaciones web del lado del servidor es una tarea compleja que los usuarios finales difícilmente pueden manejar. Este trabajo presenta un nuevo enfoque para el diseño de aplicaciones de aumentación web basado en el lado del cliente y componentes del lado del servidor. Se propone un enfoque basado en el modelo que eleva el nivel de abstracción para el desarrollo del servidor. El enfoque utiliza principios avanzados de separación de conceptos. Por lo que se proporciona un conjunto de herramientas para diseñar la composición del núcleo de la aplicación y el aumento. Las ideas presentadas y la propuesta se ilustran con varios ejemplos corrientes que muestran el potencial del enfoque.

1. Introducción

Las aplicaciones web a menudo se diseñan y desarrollan teniendo en cuenta un pequeño subconjunto de partes interesadas (administradores, usuarios internos, propietarios de productos y desarrolladores entre otros) sin una escucha activa de la retroalimentación del público y la conciencia de sus necesidades. La aparición de la personalización de la web permitió introducir mejoras por parte de los usuarios finales, como personalizaciones o adaptaciones a una aplicación que se ejecuta como una caja negra. La personalización suele estar ligada a los sistemas de recomendación, que esencialmente sugieren artículos de información a los usuarios de acuerdo con sus perfiles de usuario [7], pero el contenido no es el único aspecto que puede ser adaptado para una aplicación. Otros enfoques adaptan la Interfaz de Usuario (UI), como el Adaptive Navigation Support definido por Brusilosky [6]. Cuando estos mecanismos son elegidos directamente por los usuarios, algunos autores los llaman personalizaciones [1], sistemas que permiten a los usuarios adaptar la forma en que la aplicación muestra su contenido y funcionalidades.

Independientemente de los mecanismos de adaptación soportados por una aplicación (personalización, customización, etc.), sus usuarios pueden tener requisitos insatisfechos. No es realista afirmar que, estando idealizado por pocas personas (por ejemplo, sus propietarios), cubre las necesidades de cada usuario. Un mecanismo popular en la actualidad aborda este problema mediante la alteración de páginas web una vez que se cargan en el lado del cliente. Cualquier página web puede ser mejorada mediante la manipulación de su DOM (Document Object Model). Es posible añadir, eliminar o cambiar contenido, estilos y funcionalidades. Esta técnica es parte de lo que se conoce como Aumentación Web [11], que es muy popular en la práctica y comúnmente se

despliega como extensiones de explorador web que empaquetan artefactos de software capaces de manipular los DOM de un rango diverso de páginas web.

En aras de la comprensión, considere el ejemplo de un artefacto de aumento simple pero real, llamado Acciones mágicas para YouTube ¹. Magic Actions es una extensión de navegador (con más de cien mil usuarios) diseñada para aumentar YouTube con funcionalidades que no están disponibles originalmente con esta aplicación, como el diseño adaptable o el control de volumen por desplazamiento de la rueda del ratón. Tenga en cuenta que, en este punto, se está introduciendo la idea de adaptar o personalizar aplicaciones web de terceros. El aumento de web tiene una comunidad interesante que combina una amplia gama de esfuerzos por las multitudes (por ejemplo, extensiones de los usuarios o de los navegadores) y por los investigadores que promueven buenas prácticas del desarrollo del software, tales como reutilización, robustez [12], etc.

La mayoría de los enfoques dependen del lado del cliente sin la necesidad de una aplicación de back-end para producir el aumento. Dicha arquitectura limita el poder del aumento porque no se beneficia de las características de colaboración (hoy en día un requisito de hecho) y los limitados recursos proporcionados por el navegador (como potencia de procesamiento o alternativas de almacenamiento).

Otros enfoques tienen una arquitectura cliente-servidor tradicional, ya que ciertos servicios no se pueden implementar sólo en el lado del cliente. Por ejemplo, la sincronización de dispositivos para soportar interfaces de usuario distribuidas [16], el uso de un sistema recomendador [42] como Diigo [13]. Todas estas contrapartes de back-end son aplicaciones dedicadas específicamente diseñadas y desplegadas para el tipo particular de aumento, pero, se desconoce la existencia de enfoques considerando ambos lados del cliente y del servidor como un enfoque de propósito general.

Una posible razón para esta falta de participación es que, en la actualidad, el usuario final sólo contribuye a diseñar mejoras en el lado del cliente, ya que no tiene los medios para participar e integrar la producción de comportamiento complejo a nivel de servidor con un alto nivel de abstracción. La falta de habilidades de codificación

¹Magic Actions for YouTube <https://goo.gl/AkFnOj>

no le permite enfrentar desafíos técnicos (por ejemplo, acceso a bases de datos, diseño y codificación de algoritmos complejos y definición de componentes de interfaces de usuario).

Las tendencias actuales en el desarrollo ágil de software [26] se basan en la creación de prototipos rápidos ya que ayuda a validar fácilmente si un nuevo requisito satisface las necesidades de los usuarios. Por ejemplo, los enfoques ágiles promueven el desarrollo incremental del Producto Mínimo Viable (MVP) [3, 35] entregado en unos Sprints considerando la contribución basada en el valor del negocio. Los enfoques de aumento de web pueden dar soporte parcial si un equipo de desarrollo desea obtener un ejemplo para mejorar la experiencia de usuario de una aplicación web o para analizar el resultado de la transgresión del contenido de otros sitios web. Mediante el uso de aumentación web, es posible realizar pruebas A/B avanzadas sin necesidad de ramificar la versión actual de una aplicación, codificar la nueva característica y crear y liberar ambas versiones para comparación.

Este documento presenta un enfoque de modelado de aumentación web contemplando una aplicación cliente-servidor que oculta la complejidad del back-end a los usuarios. Los estudios existentes demuestran que los usuarios finales son capaces de crear artefactos sencillos utilizando las tecnologías web del cliente (HTML, CSS y JavaScript), pero no hay informes similares para administrar la lógica en el lado del servidor. Un medio para reducir esta complejidad es elevar el nivel de abstracción requerido para especificar tal lógica. Con esto en mente, se propone el uso de los lenguajes de modelado web existentes, lo que también reduce el esfuerzo de desarrollo [10], en un servidor dedicado para crear cualquier complemento de back-end de aumento de web necesario. Se propone una herramienta especializada para el cliente de desarrollo de cliente que permite a los usuarios especificar y representar un modelo de la aplicación de destino (el que está aumentando) mediante la abstracción de su contenido web. Como se verá más adelante, esto puede ser un beneficio no sólo para los usuarios finales, sino también para que los propietarios de las aplicaciones tejan

sus aplicaciones y evalúen nuevas funcionalidades sin necesidad de modificar el código fuente de la aplicación.

El enfoque está dirigido a la producción de aumentos web complejos basados en mejoras del cliente, pero apoyados por artefactos del lado del servidor. Las principales contribuciones de este trabajo son: un enfoque impulsado por el modelo que apoya la producción de aumentos que beneficia de la combinación de artefactos tanto en el lado del cliente como del lado del servidor, un apoyo total para la introducción continua de este tipo de aumentos, y un proceso de desarrollo que apoya las ideas que fueron ejemplificando con la herramienta WOA[14] y la plataforma WebRatio [41].

En el trabajo titulado "A Model-driven Approach for Empowering Advance Web Augmentation - From Client-side to Server-side Support" [36], se presentan los fundamentos del enfoque y ejemplos básicos. En esta tesina, que se basa en dicha investigación, se presenta una descripción detallada del enfoque, una discusión sobre sus desafíos y beneficios, un ejemplo completo que apoya las afirmaciones y la descripción del experimento llevado a cabo.

El documento está organizado de la siguiente manera: la sección 2 describe el fondo, la sección 3 que presenta una visión general del enfoque propuesto e introduce trabajos relacionados, mostrando un ejemplo completo, la sección 4 muestra una discusión técnica sobre las dificultades que se fueron presentando a la hora de abordar la construcción de los ejemplos, la sección 5 que describe los trabajos realizados sobre la herramienta, describiendo en detalle el diseño y ejecución de un experimento y, por último, la sección 6 donde se especifican los potenciales trabajos futuros que podrían abordarse.

2. Marco teórico y estado del arte

2.1 Aumentación de aplicaciones web

El aumento de la web es un conjunto de técnicas utilizadas por un gran número de usuarios. Miles de extensiones para adaptar el contenido web se pueden encontrar en los almacenes del navegador web, y comunidades significativas apoyan algunas de estas herramientas. Los usuarios finales y otras partes interesadas con habilidades de programación pueden interactuar en dichas comunidades para la creación, el intercambio y la mejora de artefactos de aumento específicos. Por ejemplo, la comunidad Userstyles (<http://userstyles.org>) ofrece un gran número de secuencias de comandos que aumentan los sitios web al agregar más especificaciones CSS que cambian la presentación del contenido. Las comunidades de scripts de usuarios, como Greasyfork (<https://greasyfork.org/>), ofrecen repositorios de scripts con un espectro más amplio de propósitos, ya que admiten diferentes tejedores de código JavaScript (por ejemplo, GreaseMonkey o TamperMonkey). Por lo tanto, es posible cambiar no sólo el estilo sino el contenido y el comportamiento de una página web. En todas estas comunidades, independientemente de la herramienta que soporten, existe una dependencia entre usuarios con y sin habilidades de programación, ya que no todos ellos pueden implementar las soluciones que necesitan y pedir ayuda a otros. En este sentido, algunas investigaciones proponen enfoques de Desarrollo de Usuario Final (EUD) para permitir a los usuarios especificar sus propios artefactos de aumento. Estos trabajos se discuten en la sección de trabajo relacionada.

2.2 Ingeniería de enfoques orientados a modelos

En la mayoría de los enfoques de diseño web avanzados, como UWE, WebML, UWA, Hera, OOWS u OOHDm, una aplicación web está diseñada con un proceso iterativo que comprende por lo menos el modelado conceptual y de navegación. De acuerdo con el estado de la técnica de las técnicas de ingeniería web impulsadas por el modelo, estos métodos producen un modelo independiente de la implementación que puede ser posteriormente mapeado a diversas plataformas de ejecución. En aras de la claridad, nos centraremos en los modelos conceptuales, de navegación y de interfaz, ya que son bastante similares en diferentes enfoques de diseño.

2.2.1 Diseño conceptual

El modelo conceptual de una aplicación web (a.k.a. dominio de aplicación o modelo de contenido) se centra en expresar los conceptos específicos de dominio involucrados en el problema que resuelve, con sus atributos, relaciones y comportamiento asociado. Cuando se define mediante algunas metodologías, como el método de metodología de diseño de hipertexto orientada a objetos (OOHDm) o UWE, este modelo es un modelo orientado a objetos descrito con UML y compuesto de clases con sus atributos, métodos y asociaciones.

2.2.2 Diseño navegacional

El diseño de navegación de una aplicación web tiene como objetivo definir vistas, estructuras de acceso y rutas de navegación hacia los contenidos para permitir al usuario acceder y navegar fácilmente. La mayoría de los métodos de ingeniería web basan su modelo de navegación en dos primitivas de modelado: Node y Link. IFML [4] no es una excepción a esto, ya que define páginas como vistas lógicas en clases de modelo de aplicación y enlaces como la realización hipertexto de asociaciones de modelos de

aplicación que define capacidades de navegación o activación del comportamiento del sistema.

2.2.3 Diseño de interface abstracta

En OOHDM, la interfaz de usuario se especifica mediante vistas de datos abstractos (ADV). Soporta un modelo orientado a objetos para objetos de interfaz. Para cada clase de nodo, se debe definir un ADV para indicar cómo cada uno de sus atributos o subnodos debe ser presentado al usuario. En este contexto, un ADV puede ser visto como un Observer [22] del nodo, y es posible expresar sus atributos y operaciones usando un diagrama de configuración [40].

Los ADV también se usan para especificar cómo se desarrollará la interacción como resultado de eventos generados por el usuario. Estos aspectos de comportamiento se especifican mediante ADV-charts [40], una especie de diagramas de estados que representan estados y sus transiciones para un ADV dado, y son útiles cuando se necesita modelar comportamientos de interfaz ricos como el de Aplicaciones de Internet Ricas (RIA) [30]. ADV-Charts son diagramas de máquinas de estado que permiten expresar las transformaciones de la interfaz que ocurren como resultado de la interacción del usuario en un ADV determinado a través de las reglas de condición de evento.

2.3 Trabajos relacionados

2.3.1 Aumentación de aplicaciones web

El desarrollo del usuario final (EUD) fue explorado en el campo del aumento de la web para capacitar a los usuarios sin habilidades de programación para especificar sus artefactos de aumento. En un trabajo anterior, se presenta una herramienta denominada WOA (Web Object Ambient) que permite a los usuarios finales recrear un modelo de objeto de sitios web en el lado del cliente [17]. Sin embargo, puesto que el foco está en soportar complejas aplicaciones de aumento que no podrían funcionar

solo con componentes del lado del cliente, se desea que los usuarios con habilidades de modelaje sean capaces de crear la contrapartida back-end de estas ampliaciones. Este soporte generalizado de back-end potenciará los artefactos de aumento ya que podrían contemplarse otras características, tales como sincronización de datos, lógica de negocio compleja, almacenamiento o aspectos sociales. Varios aspectos sobre el aumento se han abordado a través de actividades de modelado, como la especificación de requisitos [14]. Estos aspectos suelen ser dirigidos a modelar la capa de presentación, dejando de lado la especificación de comportamiento que puede tomar ventaja de los componentes del servidor. Tales modelos no son suficientes para representar la lógica de back-end de una aplicación. En este sentido, este trabajo propone la integración del enfoque para extraer un objeto modelo de las páginas web existentes [17] y una aplicación back-end creada utilizando los lenguajes de modelado web existentes. La aumentación web podría ser considerada como un mecanismo de aplicación extranjera, ya que puede adaptar los sitios web de terceros en el lado del cliente. Sin embargo, las adaptaciones de cliente-lado también podría ser visto como una preocupación central durante el diseño de la aplicación. Un enfoque interesante bajo este enfoque utilizó dicho mecanismo en un sistema de e-learning [8]. Propone una capa de modelado para la adaptación al cliente, pero no se utiliza desde el punto de vista del aumento y su uso está restringido a los propietarios de las aplicaciones.

2.3.2 Separación de conceptos en MDWE

Varios enfoques de la ingeniería web dirigida por modelos (MDWE) permiten componer sin problemas las concerns de las aplicaciones web, tales como en el trabajo anterior dirigido a abordar Funcionalidades Volátiles en Aplicaciones Web [39, 20], la separación de modelado de concerns en Workflows [38], o la separación de las concerns de Web-GIS en aplicaciones web [37]. Otros enfoques soportan requisitos evolutivos, como el Modelo de Proceso de Composición Web [21], el Distributed Concern Delivery [9] o el apoyo a principios más generales, como refactoring [19] y patrones de diseño [22]. Sin embargo, todos estos enfoques se centran en la composición de las concerns en

el lado del servidor y sólo cuando el desarrollador tiene acceso al núcleo de la aplicación. Este escenario no es el objetivo de la aumentación web.

2.4 Enfoque de aumento web impulsado por modelos

En esta sección, se presenta una breve caracterización de los requisitos de aumentos que exigen soporte de cliente y servidor. Y se presenta el enfoque donde cada uno de sus pasos se explora en detalle.

2.4.1 Justificación sobre el uso del enfoque

La práctica más común es agregar esta capa de aumento a las aplicaciones web existentes en el lado del cliente. Como ya se ha mencionado en este documento, esto se logra manipulando el DOM de las páginas web una vez que se cargan en el navegador web. Como el lector sabe en este punto, esto se logra utilizando artefactos de software independientes que los usuarios finales manejan en sus dispositivos (de escritorio o móviles) navegadores web. Este trabajo propone explorar cómo apoyar el efecto de aumento a través de un servidor back-end. El soporte del lado del servidor para el aumento de la web proviene de la necesidad de adaptar los sitios web existentes más allá de la simple manipulación del DOM. Hay varios aspectos que respaldan esta idea:

- **Alcance de ejecución:** los artefactos típicos de aumentación web [11] se ejecutan en el navegador web una vez que se carga una página web en particular. Esto limita el contexto de ejecución, dado que el artefacto de aumento, denominado aumentador, solo se ejecuta cuando se carga la página web de destino. Es posible que algunos aumentos requieran continuar trabajando incluso si el sitio web objetivo no se está utilizando, por ejemplo, porque se basa en la información que se rastrea desde otras fuentes o sitios web. Por lo general, existe la posibilidad de ejecutar un artefacto de aumento en cada página web cargada (incluso si la

página web cargada no está aumentada), y luego el aumentador podría continuar trabajando [42]. Sin embargo, este escenario todavía depende del hecho de que el navegador web debe estar abierto. En los casos en que se requiera, por ejemplo, para procesar flujos de información para realizar la capa de aumento, se requerirá un enfoque del lado del servidor para hacerlo.

- **Integración entre dispositivos:** en una estrategia del lado del cliente, los aumentadores se ejecutan en un único dispositivo, independientemente de otras ejecuciones/instalaciones del mismo amplificador en otros navegadores web. Si un usuario requiere algún tipo de integración/interacción entre sus dispositivos, entonces se necesitaría un back-end del lado del servidor. Por ejemplo, este es el caso de los aumentadores para interacciones de interfaz de usuario distribuidas [16].
- **Colaboración:** hay varias herramientas de colaboración (como Evernote, Diigo, etc.) que proponen capas de aumento basadas en patrones de colaboración, en las que los usuarios finales pueden anotar sitios web, agregar comentarios, etc. Este tipo de requisitos de aumento web obviamente necesita una base de datos común donde se almacenarán todas las contribuciones de los usuarios. Tener en cuenta a otros usuarios para las mismas capas de aumento también permite la posibilidad de ir más allá de la adaptación de UI, ya que se pueden lograr otros aspectos relacionados con la personalización, como el filtro colaborativo [24].
- **Cómputo elevado y grandes requisitos de almacenamiento:** si el aumento apunta a satisfacer requisitos que exigen una gran carga de trabajo (es decir, procesamiento de imágenes o aprendizaje profundo), el navegador web puede no ser la mejor opción para ejecutar el amplificador y, debido a los recursos limitados, la experiencia de navegación de navegación web puede verse negativamente afectada. En casos como éste, los artefactos que requieren recursos de alto rendimiento pueden implementarse en el lado del servidor y luego su resultado se entreteje en el lado del cliente.

Es importante tener en cuenta que estas cuatro dimensiones no son ejemplos de aumentadores, sino que representan diferentes concerns que pueden estar involucradas en varios tipos de requisitos de aumento.

3. El enfoque propuesto

3.1 Descripción del enfoque

El enfoque se basa en la idea de que incluso la funcionalidad más simple (por ejemplo, una nueva característica de comentario de la comunidad) debe considerarse como una funcionalidad de primera clase y, como tal, diseñada en consecuencia. Al mismo tiempo, su diseño e implementación tienen que tomarse separadas del sitio anfitrión (de ahora en adelante la aplicación principal) y lo más posible dissociadas de las funcionalidades básicas y estables que las ampliaciones no pueden ser introducidas porque el analista aumentador no forma parte del equipo de desarrollo. Sobre la base de las ideas anteriores, el enfoque se puede resumir con las siguientes directrices de diseño, que se muestran esquemáticamente en la Figura 3.1:

1. Se desacopla el aumento de la aplicación del núcleo mediante la introducción de una capa de diseño (denominada capa de aumento), que comprende un modelo conceptual, un modelo de navegación y un modelo de interfaz.
2. Se captura el modelo conceptual básico etiquetando los datos en las páginas de la aplicación del host usando un mecanismo de abstracción y estructuración [17]. La falta de acceso a los modelos subyacentes de la aplicación host requiere la extracción perceptible del modelo conceptual. En este proceso, los elementos de datos en la página son etiquetados y agrupados en una definición de entidad por un analista de aumento de tal manera que se obtiene un modelo conceptual simplificado. El analista de aumento es un experto usuario final con conocimientos avanzados de aplicación web que tiene como objetivo mejorar la aplicación principal. En otros pasos, la instanciación del modelo en una sesión de

usuario particular se utilizará para proporcionar información contextual al motor de aumento proporcionando información de instancias de modelo al activar el aumento.

3. Los requisitos de aumento se modelan utilizando las anotaciones de ingeniería web (por ejemplo, casos de uso, diagramas de interacción del usuario, etc.) y se mapean por separado en los siguientes modelos usando las heurísticas definidas por el enfoque de diseño (véase por ejemplo [32]). Se puede observar que, como se muestra en la Figura 3.2, los requisitos de aumento no están integrados en el modelo de requerimientos de núcleo, por lo tanto, dejando su integración a otras actividades de diseño.
 - (a) Los nuevos comportamientos, es decir, aquellos que pertenecen a la capa de aumento, se modelan como objetos de primera clase en el modelo conceptual de aumento. Define todos los objetos y comportamientos correspondientes a los nuevos requisitos. Además, este modelo puede incluir clases conceptuales de aplicaciones centrales (capturadas en el Paso 2) percibibles por el usuario final que permiten definir relaciones entre el modelo de negocio de aumento y la aplicación principal. Observe que esta estrategia puede aplicarse a cualquier método orientado a objetos, es decir, a cualquier método que utilice un enfoque de especificación similar a UML.
 - (b) Los nodos y los enlaces pertenecientes al modelo de navegación de aumento pueden o no tener vínculos con el modelo de navegación principal. El modelo de navegación central también es ajeno a las clases de navegación de aumento, es decir, no hay enlaces u otras referencias desde el núcleo a la capa de aumento. Este principio se puede aplicar en cualquier enfoque de diseño web.
 - (c) Se utiliza una especificación de integración independiente para especificar la conexión entre nodos de núcleo y de aumento. Como se muestra más adelante en el documento, la integración se logra en tiempo de ejecución como parte de un motor de tejido del lado del cliente. En otros enfoques

basados en modelos, la integración puede realizarse durante la transformación del modelo implementando las transformaciones correspondientes.

- (d) Se diseñan (e implementan) las interfaces correspondientes a cada concern (núcleo y aumento) por separado. El diseño de la interfaz de las clases núcleo (descrito en OOHDM utilizando Vistas Abstractas de Datos (ADV) [40]) son inconscientes con respecto a la interfaz de concerns de aumento. Como en la capa de navegación, este principio es independiente del enfoque de diseño.
4. Las interfaces de núcleo y de aumento (en las capas ADV y de implementación) se tejen ejecutando una especificación de integración, que se realiza mediante transformaciones DOM. Una vez más, la idea de tejido modelo es genérica y por lo tanto el mismo resultado se puede obtener utilizando otra solución técnica.

Una vez que los requisitos de aumento son modelados en el Paso 3, el aumento de otro sitio será bastante sencillo, requiriendo encontrar y mapear las clases virtuales (Paso 2) y especificar cómo activar los artefactos de UI de aumentación (Paso 4).

A continuación, se explica cómo los principios antes mencionados se han puesto en práctica a través de los enfoques WOA e IFML. El código fuente de la herramienta WOA y las pautas para configurar el entorno están disponibles en un repositorio git ¹.

3.1.1 Extraer el modelo conceptual de la aplicación principal (Paso 2)

El enfoque se basa en la creación de objetos que se especifican mediante la abstracción de contenidos de páginas web [15].

Para ello, se ha desarrollado una herramienta de programación visual que permite a los usuarios seleccionar un elemento DOM desde el que se inicia el proceso de abstracción. Para este elemento DOM, el usuario debe definir cuál es la clase conceptual, que son las propiedades de esa clase y también seleccionar de qué elementos DOM de los hijos

¹<https://bitbucket.org/pmaglione/mdwaplantingplanning>

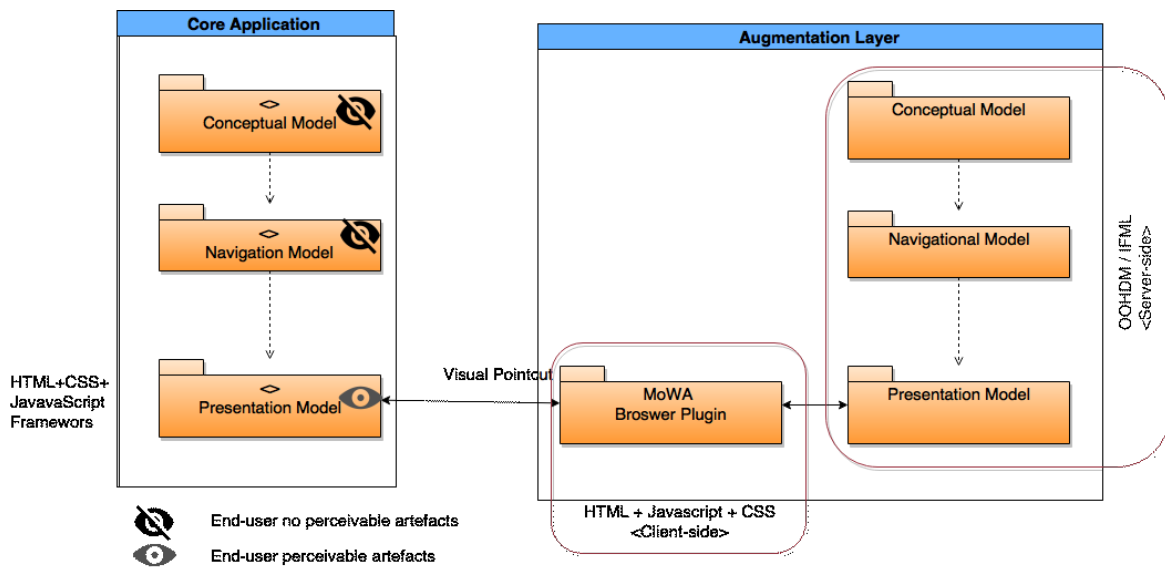


Fig. 3.1 Esquema del enfoque

son los valores de estas propiedades tomadas. Este proceso se muestra en la Figura 3.2, la herramienta añade los controles necesarios que permiten a los usuarios crear objetos, independientemente del recurso web que se haya cargado en el navegador.

El primer paso es habilitar la selección DOM (Paso 1). Al hacer clic en esta opción, cada elemento DOM se resalta en un evento de mouse-over, por lo que el usuario puede apreciar claramente cuál es el elemento de destino actual para recopilar. Entonces, como se muestra en el paso 2, puede acceder a través de un menú contextual a las opciones para extraer un elemento en el DOM actual. El paso 3, una vez seleccionado el elemento DOM, se abre un formulario de interfaz de usuario en la barra lateral, que permite al usuario seleccionar un nombre para el concepto, una etiqueta semántica, etc.

A continuación se presentan los 3 pasos de este proceso, que se pueden aplicar a cualquier sitio, a través de un ejemplo, tomando el sitio "ArgAgro" como la aplicación central. Habilitar la selección de DOM (1) implica hacer clic en un botón en la barra de herramientas del navegador. Tal interacción permite seleccionar el elemento DOM, que aplica una clase especial de CSS para resaltar el elemento del DOM actualmente apuntado, por lo que el usuario puede ayudar a identificar claramente el elemento

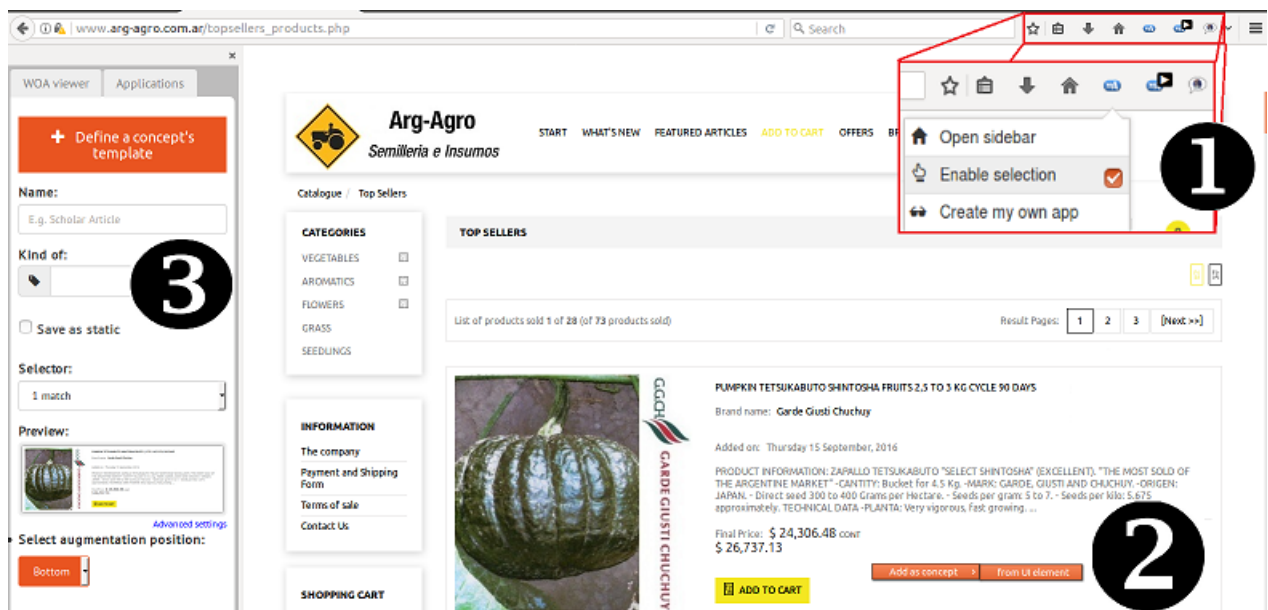


Fig. 3.2 Definición del concepto

objetivo real que se puede recolectar con la herramienta. Como resultado de esta interacción, se habilita un menú contextual especial, que permite definir un elemento concreto en el DOM como un concepto o como una propiedad (2). Una vez que se seleccionó el elemento DOM, se abre un formulario de interfaz de usuario en la barra lateral (3), que permite al usuario seleccionar:

1. un nombre para identificar el concepto. Por ejemplo, "Semilla de ArgAgro".
2. una clase conceptual para asociar. por ejemplo, "Semilla".
3. la cantidad de ocurrencias que se pueden extraer de una sola clase. Esto es transparente para el usuario, solo se puede combinar con las ocurrencias, pero internamente se trata de un selector de XPath que coincide con uno o varios elementos de la UI. Por ejemplo, `//*[@class='product-container']` en dicha página se puede usar para recuperar de 1 a 28 instancias.
4. una posición para el aumento en la relación con el elemento UI que coincide con el concepto, por ejemplo, "arriba" o "abajo".

En cuanto a las ocurrencias, un combo se llena automáticamente con múltiples opciones que coinciden con diferentes XPath's aplicables al elemento DOM seleccionado. Es posible hacer una referencia inequívoca a dicho elemento individual, o a un conjunto de elementos similares en su lugar (por ejemplo, los que tienen el mismo conjunto de clases CSS, los que están en el mismo nivel en el árbol DOM). El usuario debe elegir las ocurrencias de acuerdo a sus necesidades.

Las propiedades se pueden recoger de la misma manera, la única diferencia es la adición de un combo para vincular dicha propiedad a cualquiera de los conceptos existentes definidos para la aplicación web actual.

Al final de esta etapa, los usuarios pueden ver las clases y las instancias recopiladas en un panel especial en la barra lateral, desde donde el usuario también puede exportar las especificaciones en formato JSON. Otros aspectos de la abstracción del contenido web como objetos de dominio se pueden encontrar en trabajos anteriores [17].

El resultado de la extracción del modelo conceptual es la definición del "modelo conceptual percibido por el usuario" y la "configuración de aumento" (por ejemplo, dónde se colocará). Ambas especificaciones se utilizarán en los siguientes pasos: el primero como punto de partida para el modelado de capa de aumento, el último para crear una instancia de los conceptos concretos en tiempo de ejecución.

3.2 Modelado de la capa de aumentación

La solicitud de aumento se activa con la extensión de navegador web cuando se accede a la aplicación principal de destino y se cumplen los requisitos de ampliación, como su URL coincide con un patrón dado. La extensión pide a un servidor de aplicaciones que construya el contenido que representa el aumento e incorpora su respuesta en una posición específica de la página de la aplicación principal visitada. El servidor de aplicaciones ejecuta modelos de aplicación (Conceptual, Navegación e Interfaz) para resolver la solicitud. A continuación, se presentan las directrices para el diseño de cada modelo.

3.2.1 Modelo Conceptual (Paso 3a)

Las funcionalidades de aumento, como los sistemas de recomendación personalizados, pueden incluir clases de contenido nuevas (por ejemplo, la clase que modela un comentario de un usuario) o la mejora de las clases de aplicaciones principales existentes y su comportamiento. En el enfoque, se diseña un diagrama de clases que define nuevas clases y los enriquecimientos para las clases virtuales extraídas en el Paso 2, que se utilizarán para aumentar el comportamiento de la aplicación principal. Además, las clases virtuales surgen de los conceptos detectados en el paso anterior y son representaciones de la información realmente percibida por el analista de aumento. Las clases virtuales se mejoran con nuevos atributos (por ejemplo, un producto puede enriquecerse con un atributo de clasificación) o relaciones con otras clases virtuales o de aumento (por ejemplo, un producto puede mejorar con una relación con una clase Review). Cualquier nueva característica definida en una clase virtual será tejida automáticamente y debe ser considerada como un Decorator [22], ya que permite agregar nuevas características (propiedades y comportamientos) a una aplicación de una manera no intrusiva.

En el enfoque, las funcionalidades de aumento pueden ser nuevos comportamientos que se agregan al modelo conceptual (y que pueden abarcar muchas clases) o modelos de navegación completos, que contienen nuevos nodos, enlaces e incluso relaciones con clases conceptuales. Cada funcionalidad de aumento se trata como un subsistema autónomo y se modela utilizando el enfoque MDWE seleccionado. La notación es similar a los enfoques simétricos para la separación de concerns como la descrita en [34].

3.2.2 Diseño Navegacional (Paso 3b)

El aumento y los componentes básicos de navegación se conectan utilizando una especificación para la integración en la capa de navegación. Indica, por ejemplo, si las características de aumento están "insertadas" en el nodo central o si están conectadas

con un hipervínculo. Esta especificación también incluye una consulta indicando qué nodos principales serán considerados en la extensión. Los nodos que coinciden con la consulta se ven afectados por (o mejorados con) la funcionalidad de aumento y representan la afinidad de la funcionalidad de aumento. Es posible definir una o más afinidades para la misma funcionalidad de aumento, es decir, la misma funcionalidad podría incorporarse en diferentes partes de la aplicación, siguiendo diferentes reglas. Las afinidades de una funcionalidad de aumento se especifican con el mismo lenguaje de consulta utilizado en OOHDM para definir nodos [33] que es soportado por la herramienta, que se presenta en la siguiente sección. El lenguaje se basa en consultas de objetos. Mediante el uso de este lenguaje de consulta, la definición de una afinidad asume la siguiente forma:

AFFINITY: *AffinityName*

FROM $C_1..C_i$

WHERE *Predicate*

INTEGRATION: *Extension* | *Linkage*($V_1..V_i$)

(1)

En esta plantilla de afinidad, *AffinityName* es el nombre asociado con la afinidad, $C_1..C_i$ indica las clases de nodo núcleo involucradas en la consulta, *Predicate* es una expresión lógica definida en términos de propiedades de objetos de modelo que determina las instancias de las clases de nodo núcleo $C_1..C_i$ que serán afectadas por la funcionalidad de aumento y *Extension/Linkage* indica la forma en que la funcionalidad de aumento se integra en los nodos principales a través de los nodos de aumento $V_1..V_i$. Una extensión indica que los nodos principales se mejoran para contener la nueva información de funcionalidad (y operaciones). En una integración de enlace, los nodos del núcleo "sólo" permiten la navegación hacia los nodos de aumento $V_1..V_i$ que realmente contienen la funcionalidad de aumentación y por lo tanto no soportan nuevos comportamientos. En el caso de la integración de vinculación, también podemos especificar características adicionales como atributos o anclas que tienen que agregarse al nodo extendido (por ejemplo, para hacer la navegación más clara).

3.2.3 Enlazado Estructural de la Aumentación (Paso 3d)

Como consecuencia de la inserción de funcionalidades de aumento en el modelo conceptual o en el modelo de navegación, se deben añadir nuevos elementos de interfaz al modelo de interfaz, introduciendo nuevos campos con objetos de interfaz de datos o de control (anclajes, botones, etc.). Aunque se describe este proceso en [23], se revisa brevemente aquí por razones de integridad y legibilidad. Cada concern (núcleo y aumento) comprenderá ADVs para sus nodos de navegación correspondientes. Durante la fase de diseño de la interfaz y cuando un nodo debe mostrar alguna funcionalidad de aumento, se indica la apariencia de la página final especificando cómo se insertará la interfaz de aumento en el núcleo ADV. Más específicamente, se indica la posición relativa de los objetos de interfaz añadidos relativos a los objetos de interfaz de núcleo. Para expresar la integración, se ha definido un lenguaje de especificación simple que permite indicar puntos de corte e inserciones en el nivel de interfaz abstracto, es decir, la posición en la que el aumento de ADV tiene que ser insertado en el núcleo ADV. La especificación generaliza la idea de puntos de corte en la orientación de aspecto al espacio bidimensional de las interfaces web. Un punto de corte y su correspondiente inserción se especifican utilizando la plantilla 3.2.

`Integration: IntegrationName`

`Target: ADVTargetName`

`Add: ADVSourceName | InsertionSpecification`

`[Relative to: ADV name]` (2)

`Position:`

`[above | bottom | left | right] |`

`[float [left | center |right] [top | middle | bottom]]`

El campo *Integration* es una identificación para esta especificación. Puede referirse o no a una afinidad de navegación ya que la misma especificación de integración de interfaz de usuario (UI) se puede utilizar con muchas afinidades de navegación. El campo *Target* indica los nombres de los ADV (uno o más) que alojarán el código de interfaz de aumento. Los ADV internos se pueden especificar usando un punto. Como

ejemplo, escribiremos *Product.Reviews* para indicar que la inserción se llevará a cabo en las revisiones de ADV, que es una parte de ADV del producto. El campo *Add* indica qué elementos deben insertarse en el destino, ya sea un ADV o una especificación inmediata que se utiliza cuando el campo insertado es lo suficientemente simple para evitar la especificación de otro ADV (auxiliar). Finalmente, se indica la posición de inserción usando los campos *Relative* y *Position*. Cabe señalar que la especificación sigue siendo "abstracta", lo que permite el ajuste fino para la siguiente etapa, durante la implementación.

3.2.4 Implementación en la nube

Hasta ahora, se han diseñado concerns de aumento que promueven soluciones que se destacan debido al acoplamiento débil entre la aplicación del núcleo y la capa del aumento que promueven la reutilización de los activos del aumento. Se propone una forma de afrontar la reutilización, basada en un catálogo de scripts de aumento, artefactos de fondo y documentación de uso, definidos de una manera muy abstracta e independiente del sistema. La idea es usar estos incrementos documentados ahorrando tiempo y esfuerzo al mejorar las aplicaciones. El catálogo se alimenta con aumentos a medida que son analizados y diseñados por los analistas de aumento. Sin embargo, estas ampliaciones no son estáticas y pueden evolucionar con el tiempo, cuando se reutilizan en otras aplicaciones y también de acuerdo con la retroalimentación de los usuarios. En la figura 3.3, se presenta el esquema de implementación de dos incrementos diferentes publicados en la nube. En la ilustración, un navegador web está ejecutando la *aumentación 1*, que tiene su propio servidor de aumento responsable de proporcionar activos y lógica de negocio para mejorar la aplicación. Además, hay un segundo navegador web que ejecuta dos diferentes aumentos (1 y 2) que dependen de su propio backend. Finalmente, hay un componente de catálogo en esta arquitectura que tiene referencias a los activos de aumento.

Cada nuevo aumento se describe en el catálogo utilizando una plantilla simple compuesta de los siguientes campos:

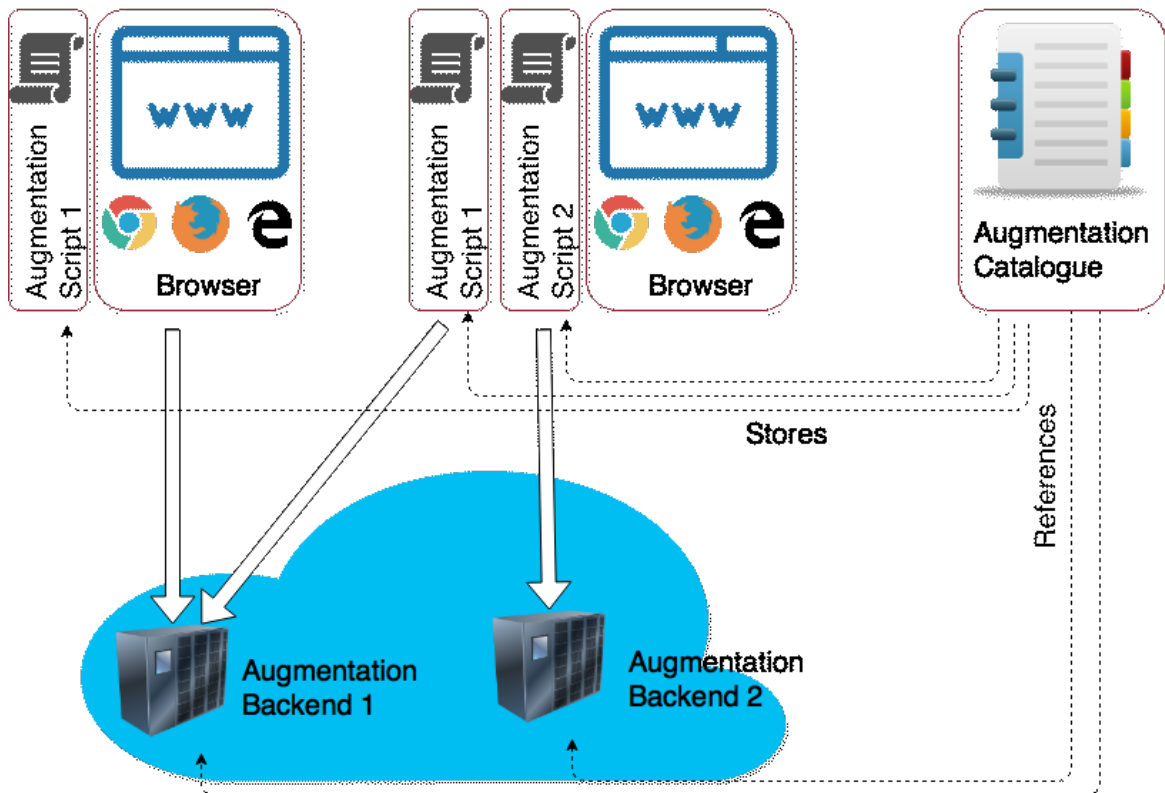


Fig. 3.3 Esquema de implementación en la nube

1. **Nombre:** un nombre corto que describe el aumento.
2. **Descripcion:** una breve descripción del comportamiento de aumento.
3. **Referencias:** una lista de aplicaciones web donde el aumento se ha introducido con éxito.
4. **Lado del cliente:**
 - (a) **Modelo Conceptual:** se espera que el modelo conceptual sea alimentado usando la información percibida en la aplicación objetivo. Debe estar claramente definido para una comprensión apropiada del aumento actual por los analistas de aumento. Cuando quieren reutilizar el aumento, comprobarán si un nuevo sitio objetivo puede proporcionar un modelo conceptual similar.

Si no se satisface este modelo conceptual, no hay garantía de que la lógica en el backend funcione como se esperaba.

- (b) **Configuración de aumento:** la configuración de aumento, que ha sido definida por un usuario final, permite ejecutar el aumento en una aplicación web conocida. Los usuarios pueden reutilizar esta especificación para ahorrar tiempo.
5. **Diseño de aplicaciones del lado del servidor:** los diagramas de diseño de una aplicación web (utilizando cualquier enfoque MDWE, como IFML) se adjuntan con el fin de describir el comportamiento de backend.
 6. **Capturas:** las capturas de pantalla de la instanciación del aumento deben adjuntarse para representar la experiencia resultante.

Además, los problemas conocidos pueden agregar valor al catálogo. El primero describirá las excepciones y limitaciones en el aparato del concern, mientras que el segundo describirá el impacto en la aplicación de destino.

3.2.4.1 Herramienta de Catálogo

Desarrollé una herramienta basada en el framework Symfony2 simple pero efectiva para registrar y documentar la información del aumento. Para ilustrar el uso del catálogo, se muestra en la figura 3.4 una lista de aumentos y su información requerida disponible en el repositorio. Una vez que el usuario selecciona un aumento de la lista, se le ofrece una página con sus detalles. En el ejemplo presentado en la Figura 3.5, el lector puede apreciar información relevante como el nombre, descripción y referencias, así como otros detalles relacionados con el aumento. Todos los activos necesarios para ejecutar el aumento están disponibles en línea, como las configuraciones, las secuencias de comandos del navegador web y los modelos del backend.

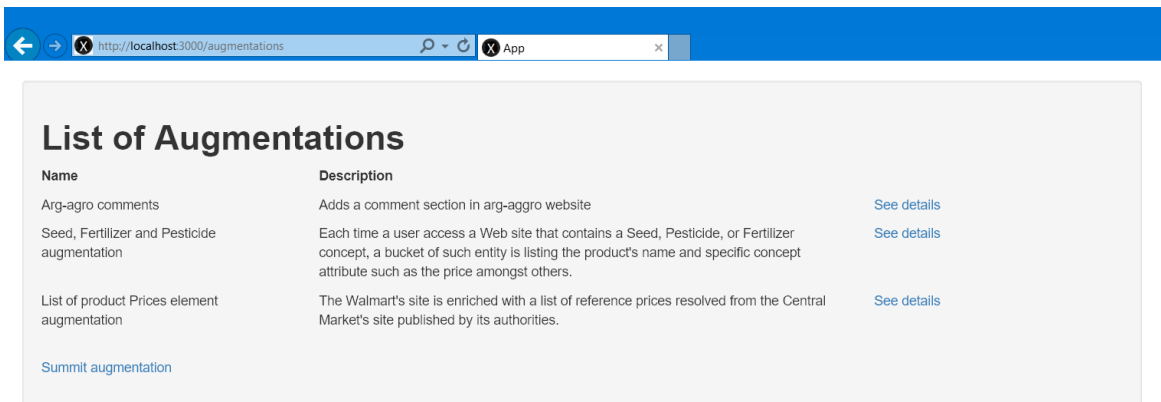


Fig. 3.4 Una lista de los aumentos disponibles en el catálogo

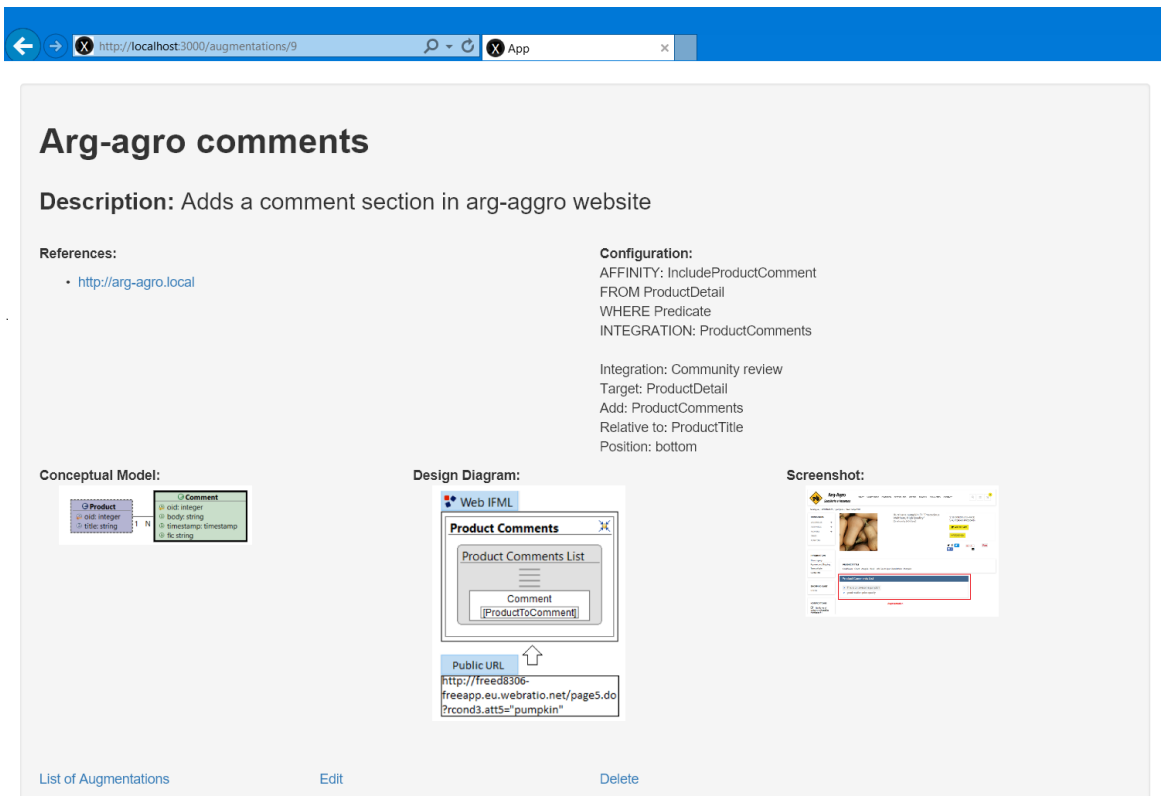


Fig. 3.5 Un aumento almacenado en el catálogo

3.2.5 Aplicación de plantillas de extracción: soporte del lado del servidor y del cliente

3.2.5.1 Hacia una navegación con estado

Como se explica, el contenido del aumento es generado por los componentes del servidor y se procesa en el lado del cliente con una extensión instalada en el navegador.

Para ello, una capa conceptual se define anotando elementos DOM y generando plantillas de extracción, como se muestra en la figura 3.2. En el enfoque, cuando el componente cliente detecta que la página web cargada coincide con una plantilla de extracción en particular, extrae la instancia del concepto actual, solicita al servidor la capa de aumento y teje esta capa en la página web actual. La capa de aumento se construye utilizando las instancias extraídas anteriores. La figura 3.6 muestra esta idea. A la izquierda, el usuario ha visitado algunos sitios web y ha extraído las siguientes clases: lechuga (A), pimiento rojo (B) y sandía (C). Estas tres instancias se almacenan en Server-Side, y el objetivo es aumentar un sitio con una nueva sección que presenta información relacionada con las instancias previamente visitadas, como a la derecha de la figura 3.6, donde los tres conceptos son Mostrados juntos (A, B y C) en la capa de aumento.

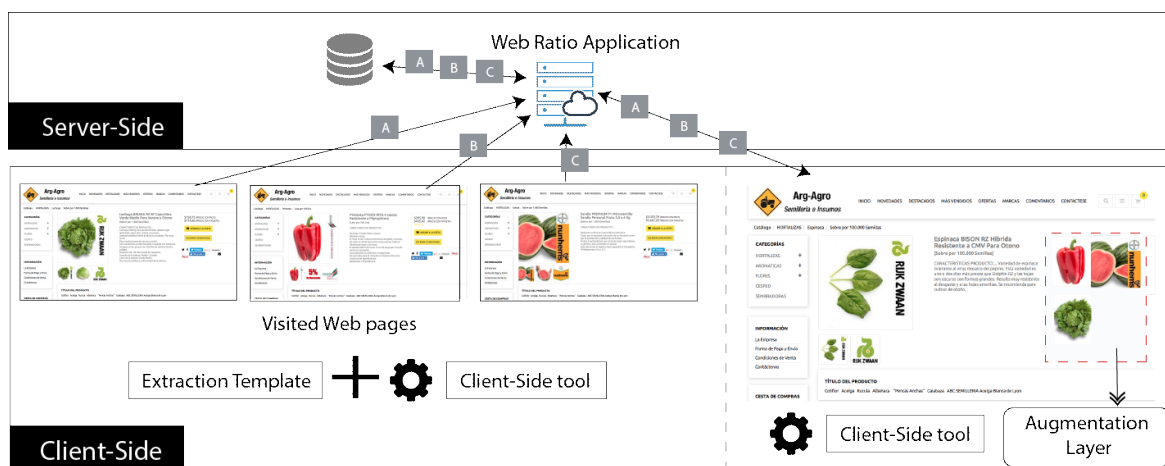


Fig. 3.6 Capa de aumento basada en instancias extraídas de sitios web visitados

3.2.5.2 API de Recuperación de instancias

En este ajuste, la plantilla de extracción se aplica en el lado del cliente y el aumento podría basarse sólo en instancias que han sido visitadas por el usuario que utiliza esta capa de aumento (aplicando la plantilla de extracción correspondiente cuando se ha cargado la página web de destino y extraer una o varias instancias). Aunque pueden satisfacerse varios tipos de requisitos de esta manera, puede ser difícil integrar la

información a petición de otros sitios web (por ejemplo, cuando la información deseada se debe recuperar al mismo tiempo que se genera la capa de aumento). Una alternativa podría ser almacenar todas las instancias de concepto en un servidor siempre y cuando el usuario visite un sitio web para su procesamiento futuro, pero la información puede quedar obsoleta. El enfoque también permite aplicar las plantillas de extracción en el backend, lo que permite a los analistas aumentar modelos de escenarios complejos, como los que requieren una recuperación dinámica de la información. La misma herramienta de modelo conceptual para definir la plantilla de extracción se puede utilizar para aplicar dichas plantillas en el backend y hacerlas públicas como un servicio.

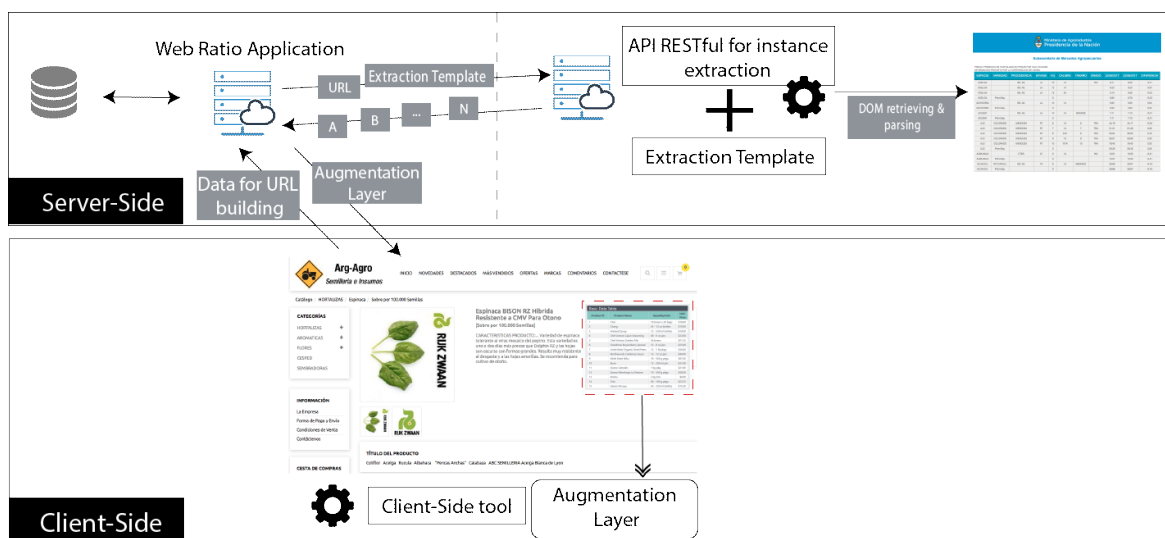


Fig. 3.7 Capa de aumento basada en instancias extraídas por IRAPI

El lector puede observar que la capa de aumento no se basa en las instancias previamente recopiladas, sino que se basa en instancias extraídas en el backend. Una aplicación implementada como servicio, que llamamos API de recuperación de instancias (IRAPI), realiza la extracción de dichas instancias. Esta API permite la recuperación dinámica de instancias mediante la descarga de una página, la extracción de un concepto (o una colección de ellos) y la producción de una carga útil que se devuelve a la persona que llama de la API. Hay dos tipos de servicios IRAPI. La más simple está configurada a priori. El desarrollador puede cargar la plantilla de extracción en el servidor IRAPI, establecer la URL desde donde se recuperará el DOM y publicar los servicios. Una

segunda opción es permitir especificar la plantilla de extracción y la URL en tiempo de ejecución, para mejorar la flexibilidad.

En el ejemplo ilustrativo de la figura 3.7, la aplicación WebRatio utiliza la plantilla de extracción y la URL como parámetros para solicitar instancias al IRAPI. El ejemplo de la figura 3.7 aumenta los detalles vegetales actuales con los precios en Argentina desde su mercado central. Para ello, WebRatio obtiene las instancias de producto de la página web del mercado a través del IRAPI. A continuación, puede procesar (por ejemplo, filtrar) estas instancias para generar el contenido que finalmente se utiliza para aumentar el sitio en el lado del cliente.

Al usar solicitudes HTTP de tipo GET, la URL debe especificarse para que el IRAPI pueda realizar las solicitudes y aumentar las páginas web con los resultados obtenidos. En el ejemplo, la consulta podría diseñarse para solicitar que el nombre del producto (concepto abstraído) se muestre en la página actual del usuario como un parámetro en la URL. Resumiendo, se admiten dos formas de obtener contenido externo: obtener información de una URL fija o generar una URL con una cadena de consulta definiendo cómo compilarla automáticamente con información contextual (desde la página web actual).

3.3 Un ejemplo comprensivo

En esta sección, se muestra y describe algunos ejemplos de aumento realizando todos los pasos propuestos en el enfoque. El objetivo es mejorar un sitio web en el dominio de la agricultura, que vende diferentes tipos de semillas, con aumentos diseñados por un analista de aumento.

3.3.1 Transmisión de contenido desde el servidor

3.3.1.1 Extraer el modelo conceptual de la aplicación principal

El primer paso en el enfoque es capturar el modelo subyacente percibido por el usuario final que se utilizará más adelante para mejorar la aplicación web. Por ejemplo, la figura 3.8 destaca una parte específica del DOM que contiene información acerca de una instanciación de la clase Product. También es posible ver que otros elementos DOM representan más instancias de la misma clase. Estas instancias parecen presentar información similar relacionada con diferentes propiedades de la clase virtual Product, como su nombre, su precio y una breve descripción. La idea principal es definir el concepto de producto como una clase, con sus propiedades y sus relaciones con los elementos DOM, por lo que es posible extraer los valores de las propiedades para cada instancia concreta.

The screenshot shows the Arg-Agro website interface. The header features the logo and navigation menu: INICIO, NOVEDADES, DESTACADOS, MÁS VENDIDOS, OFERTAS, MARCAS, COMENTARIOS, CONTACTESE. A search bar and a notification icon are also present. The main content area is titled 'NUEVOS PRODUCTOS PARA FEBRERO' and displays a grid of five product cards. The first card, 'ZAPALLO UCHIKI KURI', is highlighted with a red dashed border. Each card includes a product image, name, description, and pricing information. The sidebar on the left contains a 'CATEGORÍAS' menu and contact details for Buenos Aires, Mendoza, and WhatsApp.

Product Name	Description	Price (CONT)	Price (TARJ)
ZAPALLO UCHIKI KURI	"PIEL NARANJA TIPO HOKKAIDO PESO 1 A 2 KG	\$1.749,44	\$2.058,24
PIMIENTO CALAFYUCO INTA CALAHORRA	RESISTENTE A PHYTOPHTORA	\$811,46	\$901,69
PIMIENTO CALAFYUCO INTA CALAHORRA	RESISTENTE A PHYTOPHTORA	\$439,51	\$483,46
PIMIENTO CALAFYUCO INTA CALAHORRA	RESISTENTE A PHYTOPHTORA	\$1.397,25	\$1.536,98
PIMIENTO CALAHORRA (CABEZA DE GATO)	PARA INDUSTRIA CONSERVERA	\$774,52	\$851,97

Fig. 3.8 Identificación de la instancia y concepto en contenido web existente

3.3.2 Modelando la Capa de Aumentación

Se ha diseñado una aplicación web utilizando IFML para soportar los aspectos de aumento en el lado del servidor. Comprende los modelos Conceptual, Navegación e Interfaz. La aplicación suministra el soporte de fondo a toda la aplicación de aumento.

Una vez que las clases virtuales se extrajeron de la página, se diseñó el modelo conceptual que documenta cómo el aumento enriquece el modelo de la aplicación central. En el ejemplo, una clase virtual de productos está enriquecida con un conjunto de comentarios que representan las revisiones de la comunidad. Cabe señalar que la clase virtual modelada en IFML solo requiere un OID(Object Identification) que denota un identificador único, los atributos de la página que se manejarán en el lado del servidor y los nuevos atributos que mejoran su definición. En IFML, el primer paso es la descripción del modelo conceptual, utilizando el modelado Entidad/Relación. En la Figura 3.9, se muestra cómo un producto está asociado a sus comentarios. La clase virtual producto (con líneas punteadas) se aumenta con una referencia multivaluada a la entidad *Comment*. La referencia se denomina automáticamente *ProductToComment* por WebRatio. Es de destacar que la entidad *Comment* no necesita indicar toda la información disponible en la página, de hecho, solo se requiere un conjunto básico de información (por ejemplo, cualquier información que identifique el objeto, como un identificador o un nombre) que permita una implementación de búsqueda básica.

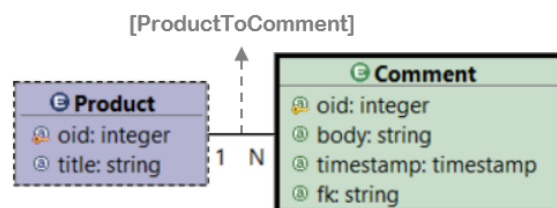


Fig. 3.9 Modelo Entidad-Relación

Las acciones en el lado del cliente requieren, además de la mejora del modelo conceptual, mejoras en los modelos de navegación para describir el comportamiento de la aplicación. Se modela utilizando unidades, componentes y enlaces como parte



Fig. 3.10 El servicio representado en Web IFML

de una vista de sitio IFML. En la figura 3.10, se define una página que lista todos los «comentarios» disponibles para un «producto» dado. Además, se necesita un identificador de «Producto» para resolver los «Comments» cuando se navega por la relación ilustrada en la figura 3.10. En la parte inferior de la figura 3.10, hay una URL responsable de desencadenar el comportamiento del lado del servidor que envía el valor requerido `id = "pumpkins"`. En realidad, la URL es solicitada por la extensión de aumento en el navegador una vez que el usuario ha visitado la página de destino.

El siguiente paso en el diseño de aumento es la especificación de cómo lograr su composición con el modelo básico de navegación de aplicaciones. Para ello, se utilizaron las afinidades introducidas en la sección 3.2.2. En la especificación 3.3, la afinidad señala que el nodo `ProductDetail` (el percibido por el analista de aumento) debe incluir el comportamiento requerido para la inclusión de los comentarios del producto, tales como la recuperación de la lista de productos y la generación de los links hipertexto necesarios.

AFFINITY: `IncludeProductComment`

FROM `ProductDetail`

WHERE `Predicate`

INTEGRATION: `ProductComments`

(3)

Finalmente, el diseñador debe estudiar el mejor diseño pensando en la experiencia de usuario para presentar la lista de comentarios. El corte visual del punto se declara en

la especificación 3.4 basado en las reglas de integración (presentado en la Sección 3.2.3). Esto determina el punto de corte de interfaz de usuario y la posición de translusión correspondiente a realizar. Cada vez que un usuario accede a una página, la instancia del modelo de aumento capturado (en este caso, un producto) se envía al servidor para su procesamiento y renderización posterior. El resultado será transluido en la página principal de la aplicación utilizando los datos de configuración. Como resultado de la ejecución de aumento en el lado del cliente, la página web se modifica listando el comentario del producto como se esperaba.

Integration: Community review

Target: ProductDetail

Add: ProductComments

(4)

Relative to: ProductTitle

Position: bottom

The screenshot shows the Arg-Agro website interface. At the top, there is a navigation menu with links for START, WHAT'S NEW, FEATURED, BESTSELLERS, OFFERS, BRANDS, COMMENTS, and CONTACT. The main content area displays the product 'Hurricane pumpkin F1 "Precocious Waltham, High Quality"' with a price of \$39,204.00 (CASH PRICE) and \$43,124.40 (PRICE CARD). Below the price, there are buttons for 'ADD TO CART' and 'WRITE REVIEW'. A 'Product Comments List' section is highlighted with a red box, containing two comments: 'This is an awesome pumpkin!' and 'good relation price-quality'. The word 'Augmentation' is written in red below the comments list.

Fig. 3.11 Resultado de la aumentación

En la figura 3.11, la sección de comentarios aumentados se insertó encima del título del producto. El tejido del contenido web se logra con la extensión Web Object

Ambient (WOA) [17], que luego llama al servicio propuesto para realizar el aumento en el cliente. Es importante destacar la importancia de un buen proceso de anotación para obtener las instancias adecuadas (en el tiempo de extracción de contenido). De lo contrario, el aumento no puede presentar ningún resultado.

3.3.3 Orquestación compleja de conceptos sensibles a la navegación

En el trabajo [18], se introduce la definición de Navigation Concerniente al Concern (CSN) para una aplicación web cuando el contenido, los enlaces y las operaciones exhibidos por las páginas de la aplicación central no son fijos para diferentes rutas de navegación, sino que pueden cambiar en el contexto de diversas concerns de navegación. Para hacer esta definición concreta y práctica se supone que los usuarios navegan a través de objetos de navegación que son la realización de nodos hipermedia. Supongamos un objeto de navegación N_j , que es una instancia de un objeto de navegación de tipo N , mientras que en la navegación web convencional este objeto mostrará el mismo contenido y enlaces independientemente de cómo se haya alcanzado, en CSN, sus propiedades pueden ser ligeramente ajustadas de acuerdo a la preocupación del usuario actual. La aumentación se puede utilizar para implementar CSN en aplicaciones web.

La herramienta [18] permite a los usuarios finales desarrollar una tarea compleja explorando los nodos de diferentes aplicaciones web. Los nodos de las aplicaciones se mejoran con el nuevo contenido, la navegación y los enlaces de acuerdo con el propósito de la tarea. Apoyando la actividad del usuario subyacente que motiva la navegación. Como resultado, las navegaciones del usuario pueden ayudar a mejorar la experiencia del usuario mientras realiza la actividad.

Consideremos ahora otro escenario en el campo de la agricultura: la planificación de una producción de hortalizas por parte de un agricultor. Una de las tareas menores de la actividad de planificación es la investigación de diferentes productos alternativos utilizados durante la producción de hortalizas. Durante esta tarea, necesita recopilar información sobre semillas, pesticidas y fertilizantes (es decir, nombre, detalles y

costo), así como preparar un informe final que resume la información de los productos preseleccionados. Por el contrario, sin un soporte de CSN, el usuario debe ser consciente de que cada vez que salta de un sitio a otro (mientras navega por diferentes sitios web), pierde la información presente en las páginas, y puede mantener notas de la información relevante para un comparación fácil. Como alternativa, hoy en día los usuarios tienden a dejar abierta una docena de pestañas con información de productos. Este tipo de práctica limita la actividad de investigación realizada por el usuario a un navegador específico porque la información no se comparte entre otros dispositivos (por ejemplo, teléfonos inteligentes, tablets y portátiles) propiedad del usuario. Para apoyar la actividad de investigación de productos en la planificación de la producción de hortalizas, se modelará CSN utilizando el enfoque, confiando en los componentes cliente y servidor.

3.3.3.1 Extraer el modelo conceptual de la aplicación principal

En la figura 3.14, se presenta el modelo conceptual compuesto por tres entidades:

- **Semilla** para capturar especies alternativas de un producto dado.
- **Pesticida** para identificar los requisitos y limitaciones de los productos y las verduras.
- **Fertilizante** para identificar el producto y la técnica utilizada.

Este modelo se utiliza para persistir en una preselección de Semillas, Plaguicidas y Fertilizantes que serán evaluados para definir qué vegetal será producido por un agricultor.

3.3.3.2 Modelado de la capa de aumento

Modelar la capa de navegación en este ejemplo es bastante similar al ejemplo de transclusión que se presenta en la sección 3.3.2 en la página 30, porque la navegación debe ser modelada y las afinidades y especificaciones de integración deben ser declaradas.

Este ejemplo presenta un desafío interesante de agregar información obtenida de la navegación de varios sitios y su persistencia. Por lo tanto, el componente de fondo desempeña un papel primario ya que permite centralizar la información de actividad del usuario almacenando el estado de navegación definido navegando en varios sitios web. Por otra parte, el componente de fondo puede utilizarse para proporcionar un entorno de computación líquido [27] y multidispositivo, donde los datos y las aplicaciones se mueven sin problemas entre múltiples dispositivos y pantallas y se sincronizan de forma transparente.

La Figura 3.14 presenta el modelo de navegación requerido para apoyar la tarea de recolección de información de los productos. Cada vez que un usuario accede a un sitio web que contiene un concepto de Semillas, Pesticidas o Fertilizantes, un cubo (Selección de Semillas, Selección de Fertilizantes y Selección de Plaguicidas) de dicha entidad enumera los detalles del producto (los atributos del concepto, como el nombre o el precio). Además, es posible agregar el producto visitado actual a la lista o eliminar un producto de la lista. En cualquier momento, el usuario final puede revisar el avance de la tarea actual accediendo a la página del informe de planificación a través de un enlace presente en cada cubo. El *flujo* de Selección de Semillas, Selección de Fertilizantes y Selección de Páginas de Plaguicidas al informe uno permite dicha navegación.

Con el concepto desarrollado en la página 30, un punto clave es la identificación y extracción de los conceptos y sus propiedades en las aplicaciones web. Esto conlleva a una secuencia de acciones de reconocimiento que recaen en los desarrolladores y en cómo compartir la información de las instancias relacionadas al modelo específico en cada sitio. Una vez que se extrae la información esta será almacenada en la API de recuperación de instancias como se detalla en la sección 3.2.5.2, la dependencia de este servicio es alta ya que el éxito en el proceso de navegación y selección de los productos depende de que los sitios visitados y la estructura de los conceptos se encuentren almacenados, otro factor es que el modelo registrado esté obsoleto y no se logre identificar las instancias correctamente. La interacción entre los componentes puede visualizarse en la figura 3.12, donde el punto de entrada al proceso es la búsqueda

de la existencia del sitio en IRAPI y la identificación de la instancia con el concepto que tiene registrado, este flujo se repetirá por cada página que el usuario acceda ya sea dentro del mismo dominio o accediendo a uno nuevo, el rol del plugin resulta vital en la interpretación de estos eventos. Habiendo existido una navegación exitosa el resultado será la visualización de las selecciones agregadas en el reporte, este ejemplo se acota al listado en dicha grilla de estos conceptos, dando lugar a nuevas posibilidades de análisis y manejo de dicha información una vez centralizada para presentarle nuevas utilidades al usuario.

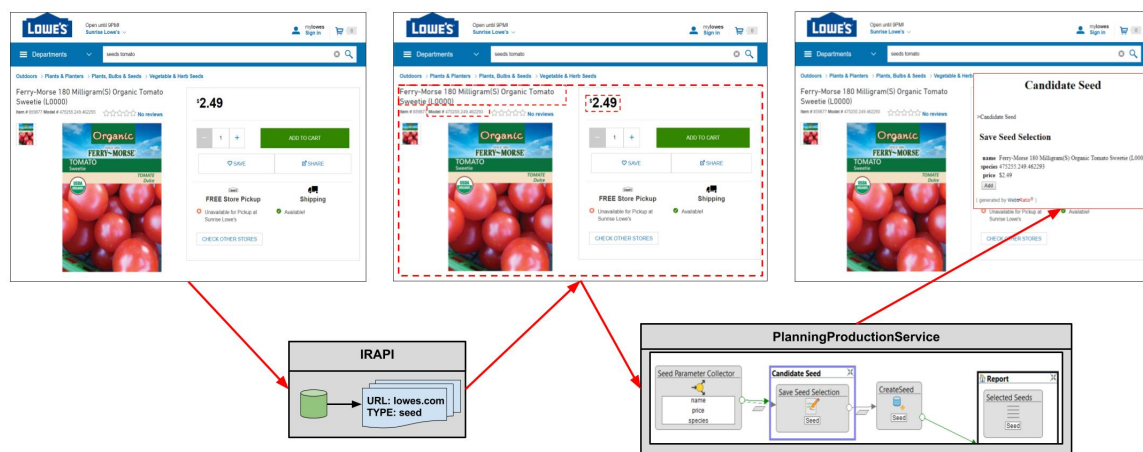


Fig. 3.12 Flujo de interacción entre componentes de la aumentación

La integración de las especificaciones que apoyan la tarea de selección de productos (Semillas, Fertilizantes, Plaguicidas) se enumeran junto a las especificaciones 3.5, 3.6 y 3.7 respectivamente:

Integration: Selecting Seed

Target: SeedDetail

Add: Selecting Seed task

Position: float right top

Integration: Selecting Fertilizer

Target: Fertilizer Detail

Add: Selecting Seed task

Position: float right top

(5)

(6)

```
Integration: Selecting Pesticide
Target: PesticideDetail
Add: Selecting Pesticide
Position: float right top
```

(7)

El resultado de enriquecer un sitio web con características que permiten al usuario recolectar información, que se utilizará más tarde, durante un proceso complejo que implica varias actividades en diferentes sitios web, se muestra en la figura 3.14. El primer paso es la selección de semillas en el sitio ² de Lowe que se aumentó considerando la especificación de integración 3.5. Entonces, varios fertilizantes son seleccionados por un usuario del granjero para ser evaluados escogiéndolos de Amazon ³ donde el aumento se basa en la especificación 3.6. Como tercer paso, se agrega un soporte para guardar una lista de pesticidas en el sitio web de TractorSupply ⁴, cumpliendo con la especificación 3.7. El usuario puede acceder a una página de resumen en la que se muestra toda la información recopilada y puede utilizarse para tomar una decisión sobre la producción del vegetal. Un aspecto a tener en cuenta es que la navegación no debe realizarse necesariamente en el orden presentado. De hecho, el usuario puede navegar por todas las aplicaciones (de ida y vuelta) hasta que obtiene la información que necesita para tomar una decisión.

3.3.4 Identificación única de los usuarios por instancia

Las características en el uso de la herramienta derivan en una necesidad de generar una identificación única para cada instancia en la que el usuario utilice la extensión de "Planificación de Producción". Se llamará instancia a cada vez que el usuario abre un nuevo navegador, contemplando que un usuario puede tener un navegador abierto si abriese uno nuevo o si cerrase el existente y lo volviese a abrir, dichos casos se tomarán como nuevas instancias. La necesidad mencionada surge para asignar

²<https://www.lowes.com>, último acceso al 27 de julio de 2017

³<https://www.amazon.com>, acceso pasado el 27 de julio de 2017

⁴<http://www.tractorsupply.com>, accedido por última vez el 27 de julio de 2017

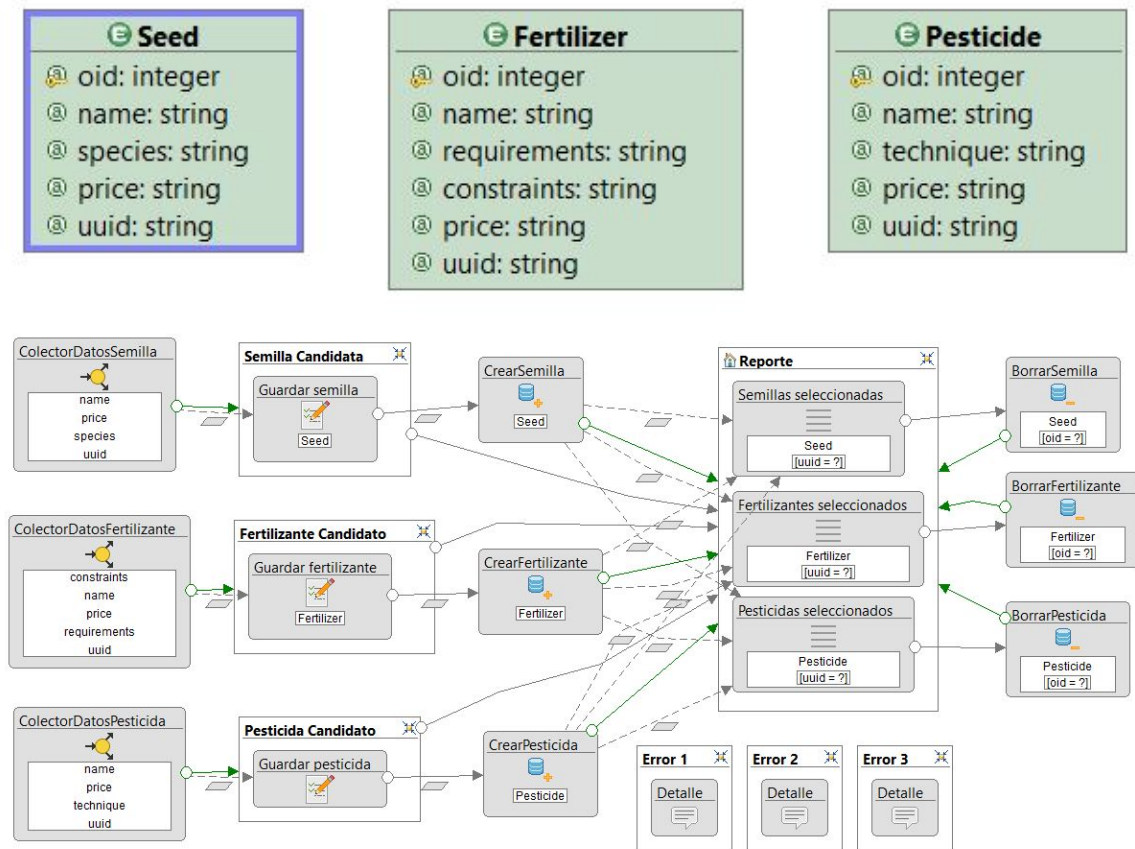
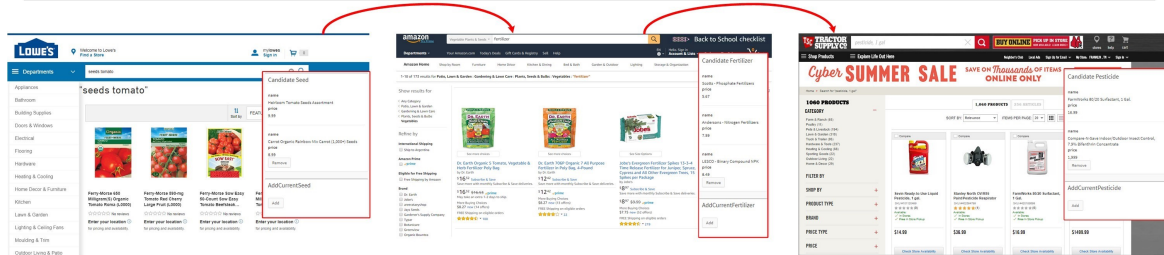


Fig. 3.13 Modelo conceptual y navegacional de la Planificación de Producción

un identificador único global a cada usuario para vincular y recuperar los conceptos que registra en su navegación. Como se puede observar en la Figura 3.14 una vez identificado el concepto el usuario puede agregarlo a su reporte con la modalidad antes mencionada, a su vez múltiples usuarios pueden estar usando la herramienta y registrando productos de su interés. Para resolver este dilema se decide utilizar un GUID. El identificador único global (GUID) es un número pseudoaleatorio empleado en aplicaciones de software. Los GUID se usan en la programación para identificar varios tipos de entidades. Es esencialmente un número de 16 bytes. Los GUID se escriben empleando una palabra de cuatro bytes, tres palabras de dos bytes y un bloque de seis bytes, aunque los dos primeros aparecen separados. Por ejemplo: 3F2504E0-4F89-11D3-9A0C-0305E82C3301. Uno de los usos es para distinguir interfaces de componentes software diferentes. Esto significa que dos versiones de un componente pueden tener

Planificación de producción



Reporte

>Reporte

Semillas seleccionadas

name	species	price
Bush Beans	Jade II Bean	\$2.60 Delete

Fertilizantes seleccionados

name	requirements	constraints	price
Purna Grow-Well 1Ltrs	Dosage: 3 - 4 ml per liter of water for foliar spray.	Perfect for Use On Indoor/Outdoor Plants.	\$599.00 Delete

Pesticidas seleccionados

name	technique	price
ORTHO Max 11.5-lb Fire Ant Killer	Ortho, 11.5 LB, Max Fire Ant Killer, Protects Entire Yard For 4 Months, Kills The Queen In 24 Hours	17.97 Delete

| generated by WebRatio® |

Fig. 3.14 Navegación y resultado en la Planificación de Producción

exactamente el mismo nombre, pero ser diferente de cara al usuario. Para el ejemplo actual en la herramienta de Planificación de Producción se utiliza una interfaz provista por el navegador Mozilla Firefox llamada nsIUUIDGenerator⁵ la cual será utilizada para generar el ID para la instancia actual del usuario en el instante que comienza su navegación utilizando la extensión. También se puede observar el modelo conceptual de la figura 3.13 contemplando en las tres entidades el identificador antes mencionado. Toda esta lógica resulta transparente para el usuario final pero es lo que permite la utilización de la herramienta por varios usuarios simultáneamente y persistente en el tiempo, sin esta metodología cada usuario podría ver los datos registrados en la historia

⁵<https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XPCOM/Reference/Interface/nsIUUIDGenerator>

pertenecientes a él mismo y los de todos los usuarios que han utilizado la herramienta ya que no existiría ninguna manera de vincular los registros con un usuario específico.

3.3.4.1 Identificación de usuario no persistente en el tiempo

El modo en que se utiliza de la técnica de identificación de instancias de usuarios y registros hace que aparezca un interrogante sobre una posible necesidad que podrían tener los usuarios, y es la de ver los datos que ha guardado en el proceso de Planificación de Producción en anteriores navegaciones. Como se ha mencionado y descrito anteriormente cada nueva instancia genera un nuevo GUID para el usuario, limitando la visibilidad de sus registros al tiempo en el que tenga abierto el navegador actual. La herramienta actualmente no aborda este caso pero es conciente de la posible necesidad, dado que el objeto de estudio del ejemplo no es implementar todas las posibles características y funcionalidades que podría tener el servicio pero si analizarlas y tenerlas en cuenta para describir el enfoque. A continuación se describe brevemente y a modo de presentación una posible solución agregando las funcionalidades de registro y el inicio de sesión de usuarios en el servicio de Planificación de Producción. Aunque WebRatio no está equipado con un módulo integrado para administrar usuarios, cuando se crea un nuevo proyecto web, el modelo de dominio ya contiene la parte que se requiere para modelar usuarios y grupos. Como puede observarse en la Figura 3.15.

El servicio proveería pantallas para ser embebidas en la UI por la herramienta, agregando la posibilidad de registro, inicio de sesión y cierre de sesión. El registro sería público para cada usuario que instale la herramienta, completando los datos básicos nombre de usuario, contraseña y email, quedaría registrado en el sistema y se utilizaría el ID auto-incremental generado al guardarlo en la base de datos en vez del GUID generado en frontend para cada instancia, para asignarlo a las instancias que guarde durante su navegación. El "inicio de sesión" es el patrón de diseño mediante el cual un usuario tiene acceso a los recursos protegidos de una aplicación. Esto normalmente se hace al validar la identidad del usuario mediante el envío de credenciales, como "Nombre de usuario" y "Contraseña". Esto abre un nuevo abanico de posibilidades para

el usuario y la aplicación del sistema en si, como por ejemplo proveerle al usuario un historial de los datos que fue registrando por fecha.

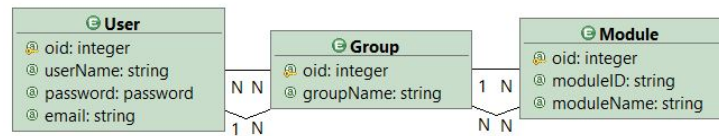


Fig. 3.15 Modelo conceptual de seguridad

3.3.5 Manejo de excepciones en la herramienta

Un punto a analizar surge de los casos en los que el flujo no es exitoso por algún motivo. Desde el punto de vista del usuario que ha instalado la extensión en su navegador surge la expectativa de que ésta interprete su navegación y presente en pantalla opciones visuales sobre las que el pueda interactuar, así como el desarrollador que utiliza la herramienta para identificar y registrar un nuevo concepto en el servidor. Como se ha presentado en la anterior subsección 3.3.3.2, la lógica que se ejecuta tanto en el navegador como en el backend está compuesta por varios pasos e interacciones cliente-servidor.

Esta sección describirá el manejo de excepciones utilizando una funcionalidad no descrita hasta el momento, la posibilidad de configurar el ambiente de la herramienta como desarrollo o producción. Esta opción determina la manera en que se manejarán las excepciones, es de esperarse que para un usuario final los errores no sean presentados en pantalla ni que estos alteren su normal navegación, es decir, que para este caso los errores deben capturarse y mostrarse en pantalla informativamente solo de ser necesario.

Para el manejo de excepciones en el ambiente de desarrollo se debe contemplar un registro de los eventos, para el seguimiento y ayuda informativa al desarrollador, se registra en la consola del navegador distintos niveles de mensajes:

- **Errores:** tagueados con el sufijo '[ERROR]: mensaje'

Table 3.1 Tabla de excepciones en en Servicio de Planificación de Producción

Excepción	Origen	Prioridad	Acción
Error en la inicialización del Plugin	Cliente	Alta	Pantalla de error N° 1
Error en la comunicación o Error interno del servidor IRAPI	Servidor	Alta	Pantalla de error N° 1
Respuesta de IRAPI: Sitio no registrado	Servidor	Media	Pantalla de error N° 2
Error en la interpretación de los atributos del concepto	Cliente	Media	Pantalla de error N° 3
Error en la comunicación o error interno del Servicio de Planificación de Producción	Servidor	Alta	Pantalla de error N° 1
Error en la inyección de la aumentación	Cliente	Alta	Alerta de error
Servicio no disponible	Servidor	Alta	Pantalla de error N° 4

- **Informativos:** tagueados con el sufijo '[INFO]: mensaje'

Los escenarios potenciales generadores de excepciones identificados se enumeran a continuación:

Una de las opciones esbozadas para este escenario es la de utilizar un diálogo de alerta⁶ por defecto presente en los DOM, no obstante esta solución termina siendo inaceptable dado que su protocolo define que es bloqueante para la pantalla y necesita una interacción del usuario para desactivarla. Este tipo de situaciones son evaluadas y evitadas en el desarrollo de los ejemplos anteriormente presentados, ya que convergen en una mala experiencia de usuario generadas no solo por los bloqueos en la interfaz sino por presentarle la obligatoriedad de realizar una acción sobre un evento que el usuario no generó.

Bajo estas condiciones, se decide establecer una pantalla de error que debe ser pedida al servicio principal en cuestión, en este caso el "Servicio de Planificación de

⁶<https://developer.mozilla.org/es/docs/Web/API/Window/alert>



Fig. 3.16 Página de error n° 1



Fig. 3.17 Página de error n° 2



Fig. 3.18 Página de error n° 3

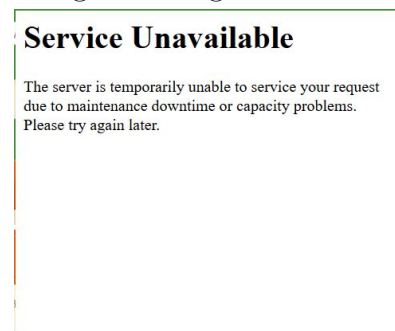


Fig. 3.19 Página de error n° 4

Producción" debe proveer un método público al cual invocar en caso de error. Dicha pantalla se insertará en la interfaz del usuario de la misma manera que se hubiera hecho con una aumentación correcta y se visualizarán mensajes de error como los que se pueden observar en las Figuras 3.16, 3.17, 3.18, 3.19, 3.20.

3.3.6 Mejora de las aplicaciones con información entre sitios

En la sección 3.2.5.2, se ha presentado un aumento que enriquece los productos de un sitio con el precio actual en el mercado central de vegetales. El ejemplo se basa en un motor API responsable de extraer los conceptos encontrados en una página. La información es requerida por medio de un API relajante expuesto que responde a una carga útil de JSON, encapsulando toda la lógica requerida para tal extracción.

A continuación, se representan los modelos y especificaciones requeridos para ejecutar el aumento. En la figura 3.21, el modelo conceptual describe los atributos más importantes: el precio mismo y la cantidad, de un *Precio*. También se presenta el

The screenshot shows the Amazon product page for 'Urban Farm Fertilizers Texas Tomato Food, Competition Tomato Fertilizer, 1 Gallon'. The page includes the Amazon logo, search bar, and navigation menu. The product image is a white jug with a red label. The price is \$28.65 plus \$11.34 shipping. The product is in stock and ships within 2 to 3 days. The page also features a list of product benefits and a 'Add to Cart' button.

amazon
Try Prime

Patio, Lawn & Garden | tomato fertilizer

Departments | Your Amazon.com | Today's Deals | Gift Cards & Registry | Sell | Help

Amazon Home | Shop by Room | Home Décor | Furniture | Kitchen & Dining | Bed & Bath

Back to search results for "tomato fertilizer"

Urban Farm Fertilizers

Urban Farm Fertilizers Texas Tomato Food, Competition Tomato Fertilizer, 1 Gallon

★★★★★ | 19 customer reviews | 5 answered questions

Price: **\$28.65** + \$11.34 shipping

In stock.
Usually ships within 2 to 3 days.
This item does not ship to **Argentina**. Please check other sellers who may ship internationally.
Ships from and sold by Urban Farms & Fertilizers.

- Hand-crafted, high-performance liquid fertilizer for all large vegetables. 4.0-2.9-6.7 NPK.
- Super-concentrated: 256:1. One gallon makes 256 gallons of full-strength nutrient. One tablespoon
- For hand-watering, drip, hose-end, foliar, soil, containers, hydroponics.
- Made from organic teas, mycorrhizae, worm casts, humic acid, kelp, enzymes, and minerals.
- Calcium rich. No more blossom-end rot.

Roll over image to zoom in

Error

>Error

Mensaje

Ocurrió un error en el Servicio de Planificación de Producción
| generated by WebRatio® |

Add to Cart

Turn on 1-Click ordering for this browser

Ship to:
Argentina

Add to List

Fig. 3.20 Página de error del servicio de planificación de producción

modelo de navegación que orquesta la llamada a la API de recuperación de información, que a su vez accede al sitio del mercado central ⁷, ⁸ para rastrear la información (*Resolver elemento de precios de mercado central*), asignar la carga útil de respuesta a la entidad de precios (*HSON / IO element*), y, finalmente, hacer la lista de precios como se esperaba (*Listado de precios del producto*). Por último el sitio Walmart ⁹ se enriquece con una lista de precios de referencia obtenidos desde el sitio del Mercado Central publicado por sus autoridades.

3.3.7 Implementación

La sección de implementación se describe en detalle en el capítulo 4, a continuación se realiza un pantallazo de los aspectos involucrados. La arquitectura se compone de dos partes. Por un lado, existe un servicio web que pretende ser público mostrando cómo interactuar con él, especificando parámetros de entrada y salida, y diseñado por WebRatio IDE para aprovechar los beneficios que aporta a la facilidad de su uso para los usuarios. Por otro lado, este servicio debe ser parte de los recursos de ampliación disponibles en el catálogo de la comunidad para hacerla más rica y crear un contexto colaborativo.

Algunas restricciones del lado del cliente se tuvieron en cuenta. Una de ellas ocurre sobre sitios que se implementan con el protocolo HTTPS(Protocolo seguro de transferencia de hipertexto)¹⁰. El intercambio de recursos de origen cruzado es un mecanismo que permite que los recursos restringidos (por ejemplo, fuentes) en una página web se soliciten invocando otro dominio fuera del dominio original. Ya que los sitios web no pueden hacer ningún tipo de solicitudes a otros dominios en la web donde se está ejecutando la extensión. Para hacer frente a este problema, se ha utilizado un

⁷http://www.minagri.gob.ar/new/0-0/programas/dma/hortalizas/01_hortalizas_act.php, Último acceso 3 de julio de 2017

⁸<http://www.alcentral.com.ar/precios.html>, Último acceso el 3 de julio de 2017

⁹<https://www.walmart.com>, Último acceso el 3 de julio de 2017

¹⁰Hypertext Transfer Protocol Secure, más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.

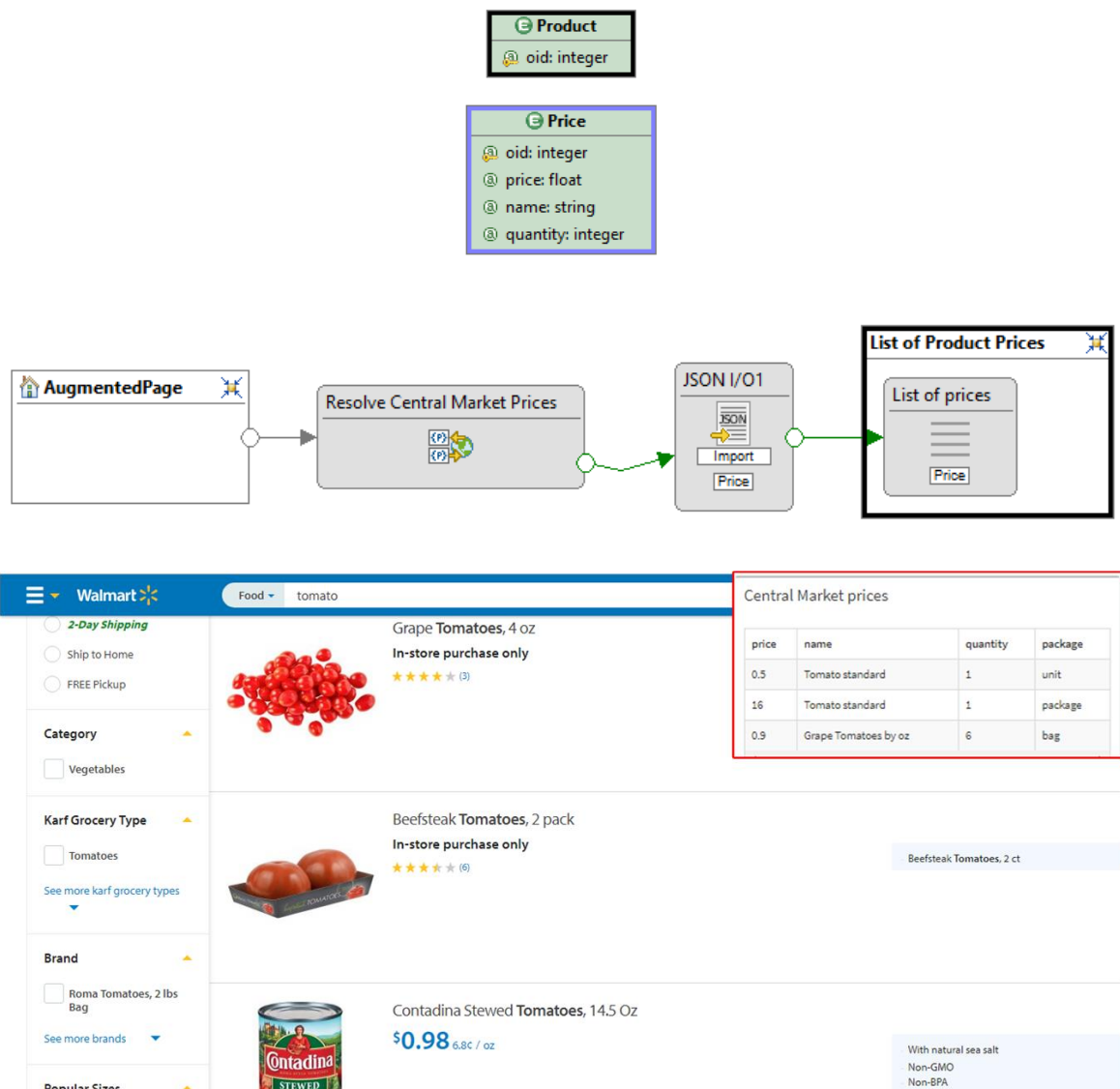


Fig. 3.21 Transluyendo los precios del mercado central con el contenido entre sitios

proxy, que permite las llamadas de dominio cruzado por envolverlos y la inserción de los encabezados de solicitud. De esta manera, se pueden hacer llamadas para traer datos como servicios de aumento. Además, el script está programado en JavaScript puro para evitar dependencias de cualquier biblioteca del lado del cliente, como podría ser el caso de jQuery ¹¹.

3.3.8 Cómo se realiza el intercambio de datos entre el frontend y el backend

Uno de los pilares principales del enfoque presentado es el soporte de backend de aumentos web. Para que esto sea posible, la herramienta debe permitir el manejo transparente de las solicitudes desencadenadas desde los aumentos a los componentes de servicios de fondo, y la detección del estado de ejecución para realizar un manejo adecuado, ya sea en caso de éxito o fracaso. La infraestructura maneja esta actividad y el analista de aumento o el usuario final no requieren una acción explícita.

Para interactuar con los servicios de backend se utiliza un objeto JavaScript llamado XMLHttpRequest ¹². Como XMLHttpRequest regular, la especificación de este objeto define una API que permite al cliente comunicarse y transferir datos. Desde el lado del cliente, realizó solicitudes y usó la respuesta de los componentes del lado del servidor para adaptar la página web.

Como alternativa a la herramienta WOA, el analista puede usar una estrategia de aumento basada en proxy [2] para inyectar un aumento basado en Javascript a las páginas web, evitando que la aplicación del lado del cliente haga llamadas ilimitadas a diferentes dominios. Esto tiene como objetivo permitir solicitudes de dominios cruzados al servicio de backend, evitando la "misma política de origen" ¹³. Esta política restringe

¹¹jQuery es una biblioteca de JavaScript multiplataforma diseñada para simplificar las secuencias de comandos del lado del cliente de HTML

¹²https://developer.mozilla.org/en-US/docs/Archive/Add-ons/Add-on_SDK/Low-Level_APIs/net_xhr

¹³https://www.w3.org/Security/wiki/Same_Origin_Policy

la interacción entre dos orígenes diferentes, con el objetivo de evitar la ejecución de scripts maliciosos.

CORS describe los encabezados HTTP que proporcionan un mecanismo para solicitar contenido de URL remotas cuando tienen los permisos para la comunicación entre dominios. Aunque el servidor puede realizar alguna validación y autorización, generalmente es responsabilidad del navegador aceptar esos encabezados e imponer las restricciones adecuadas.

A medida que el navegador realiza la detección de CORS a través de los encabezados de las solicitudes, se usa un mecanismo que es responsable de manejar correctamente los encabezados, lo que permite la ejecución de llamadas sin ser bloqueadas. El "CORS Anywhere"¹⁴ juega el papel de un proxy que agrega los encabezados CORS adecuados a las solicitudes. Esta biblioteca se proporciona como un servicio público abierto¹⁵, que puede ser libremente – y limitado – accedido. El servicio requiere una cadena de consulta con una URL como parámetro. La figura 3.22 representa el manejo de la solicitud del ejemplo "google.com" que se envía como parámetro en la URL. Una vez que el servicio recibe la solicitud, se reenvía a la URL de destino, y la respuesta finalmente se envía al script de aumento con encabezados CORS válidos. De esta forma, cualquier aplicación puede usar una solicitud XMLHttpRequest en una página para interactuar con aplicaciones de otros dominios.

3.4 Evaluando el enfoque

Este trabajo promueve el manejo de aumento en aplicaciones web como requisitos de primera clase y propone una estrategia para diseñar, implementar y desplegar apropiadamente los aumentos correspondientes. Como las aplicaciones web comprenden varios artefactos diseñados e implementados con diferentes tipos de herramientas, el impacto de la introducción de incrementos plantea diferentes desafíos para tratar con diversas tecnologías y conceptos. Con el fin de validar el enfoque y evaluar sus

¹⁴<https://github.com/Rob--W/cors-anywhere>

¹⁵<https://cors-anywhere.herokuapp.com/>

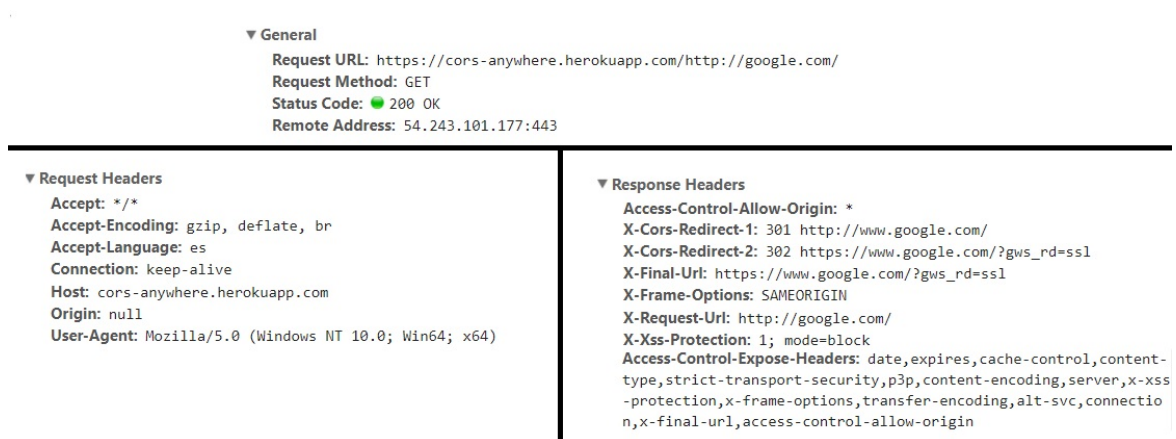


Fig. 3.22 Ejemplo de solicitud utilizando el servicio CORS

beneficios, se aplicó para aumentar una aplicación de comercio electrónico utilizado como ejemplo. El sitio fue enriquecido con el estado de comportamiento, los requisitos de CSN y sofisticadas integraciones de contenido a través de la IRAPI.

Se ha aplicado previamente el enfoque utilizando la herramienta para capturar los principales conceptos de aplicación (Paso 2) e implementar el comportamiento de backend con WebRatio (Paso 3). Ahora, se realiza un análisis de impacto de alto nivel de las principales ventajas y desafíos de las diferentes alternativas para implementar aumentos web que comprenden características de cliente y servidor.

3.4.1 Impacto

Para cada capa de aplicación web, se esbozan los cambios potenciales que podrían introducirse al implementar un requisito de incremento mediante: 1) usar un enfoque orientado a objetos convencional; 2) una técnica de aumento estándar con componentes cliente y servidor; y 3) usar el enfoque propuesto.

- **Ad-hoc:** en este contexto, las ampliaciones se gestionan como un nuevo requisito, por lo que el código fuente de la aplicación se modifica respetando su proceso de desarrollo (ágil o RUP). Cuando se utiliza un enfoque "orientado a objetos" (OO) para implementar el aumento, nuevos elementos como variables, relaciones y accesorios deben codificarse dentro de las clases existentes que los analistas

de aumento no tienen acceso en absoluto. Los cambios comprometen todas las capas de aplicaciones web: conceptual, navegación e interfaz. En este caso, las extensiones son nuevas características que tienen acceso completo a los recursos de la aplicación, como conexión a bases de datos, recursos estáticos, procesos empresariales, etc. Sin embargo, el ciclo de vida del cambio debe gestionarse manualmente, incluyendo desarrollo, pruebas e implementación. Tareas de la nueva función. Además, si la característica ya no es necesaria, debe quitarse del código fuente, la aplicación debe ser probada de nuevo y finalmente liberada. Los usuarios finales sólo pueden contribuir cuando participan en el proceso de desarrollo como parte de los estudios de la experiencia del usuario y no colaboran en el proceso de desarrollo de software. Cuando se utiliza un enfoque OO tradicional, el desarrollo de la funcionalidad volátil no sólo compromete la tarea de codificación, sino que también exige esfuerzo adicional en las pruebas, ya que las características afectadas deben ser probadas dos veces, tanto para la introducción como para la eliminación del requisito de aumento, No la adopta, como aumentos. El mantenimiento del código volátil ha sido estudiado en otros trabajos anteriores, donde se señala las consecuencias perjudiciales de mantener este tipo de características.

- **Aumento web del lado del cliente (CS Aug.):** en este caso, los usuarios finales cualificados con un sólido conocimiento de la programación en tecnologías de vanguardia como HTML, CSS, frameworks de JavaScript y desarrollo de extensiones de exploradores web, emprenden el desarrollo de la mejora de aplicaciones web modificando la estructura del DOM. El resultado de esta tarea es un conjunto de recursos JS empaquetados en un complemento de navegador web (es decir, las extensiones de Chrome o las extensiones de plugin de GreaseMonkey ¹⁶) vinculadas a la aplicación. No obstante, los aumentos se limitan a las extensiones del navegador y al contenido disponible en Internet, que puede ser eliminado para producir nuevos contenidos. Cuando ya no es necesario un aumento, el

¹⁶<http://www.greasespot.net>

usuario simplemente desactiva el script y el aumento deja de funcionar. Los equipos de Experiencia de Usuario pueden beneficiarse de esta característica para diseñar mejoras de aplicaciones, pruebas y finalmente promoverlas como características principales al fusionar los artefactos de aumento con el código fuente de la aplicación.

- **Cliente basado en código y Aumentación Web en el servidor (CB C&S Aug.):** Como un cliente regular y el lado del servidor de aumento de las demandas, como se dijo en el caso anterior, desarrollador altamente calificado capaz de entender las tecnologías de cliente y servidor (es decir, CSS, HTML, AJAX, Java, Python, etc), así como protocolos HTTP para implementar un aumento. Aunque existe una fuerte comunidad de aumento de web, este enfoque basado en código excluye a casi todos los usuarios finales debido a los conocimientos técnicos necesarios. A menudo, las soluciones de bajo nivel reducen el nivel de abstracción y los aumentos se convierten en artesanías con una capacidad de reutilización limitada. En general, uno de los principales inconvenientes del uso de enfoques convencionales (por ejemplo, aproximaciones OO) para manejar concerns de aumento es el hecho de que su ciclo de vida tiene que ser soportado de una manera ad hoc, mediante codificación en cadena de la lógica asociada. Eso no es posible en el contexto de aumentación web debido a su naturaleza: los usuarios mejoran sitios que buscan un nuevo objetivo no soportado.
- **Aumentación Web impulsada por el Modelo (MDWA):** el enfoque es apoyado por técnicas impulsadas por modelos evitando la codificación de cualquier componente, además los modelos son auto documentados, por lo tanto, la documentación de los aumentos es simple. El comportamiento de aumento se limita a los recursos manejados por la capa de aumento. Es decir, el aumento sólo puede administrar (instanciar y modificar) los objetos definidos en el modelo de aumento, y sólo puede reutilizar el comportamiento percibido accesible desde la página. No es consciente de otras rutinas básicas subyacentes. Las clases se

modelan en un nuevo paquete UML que está aislado de la aplicación de destino. Cualquier mejora de clase, como variables y métodos, se modelan en un decorador de clase de destino que luego se compone con el concepto de núcleo mediante la extensión del explorador web. Desde el punto de vista de navegación, la adición de nuevos nodos en la capa de navegación no introduce ninguna modificación de la aplicación principal porque se definen de forma independiente en su propio ámbito y posteriormente se tejen mediante la extensión del navegador web. Por lo tanto, el enfoque promueve el empaquetado de nuevos nodos de tal manera que sean fácilmente enchufados y desconectados. Con respecto a la capa de interfaz de usuario, se especifican nuevas características estructurales (como widgets, configuración de diseño o composiciones complejas de elementos de la interfaz de usuario) que posteriormente se tejen mediante un motor de transformación. Esta técnica evita el desarrollo de código UI complejo dentro del código de la aplicación principal (que el usuario final no tiene acceso y, con toda probabilidad, no podrá codificar porque no es un programador). La interfaz de usuario base permanece sin cambios en el código fuente. La gestión de este tipo de aumento de la web es bastante sencilla porque, con el enfoque, el comportamiento básico no se modifica y, por lo tanto, la prueba de regresión no es necesaria. Cuando la funcionalidad de aumento ya no es necesaria, sólo se desactiva en el navegador web. Finalmente, la productividad debe ser mejor que los enfoques basados en el código apoyados en la productividad del 900 exigida a WebRatio, presentada en [10], que aunque no sea un documento académico (todavía no hay evaluaciones de rendimiento académico en IFML), es un resultado basado en un gran número de proyectos.

En la tabla 3.2 se destacan las ventajas y desventajas de cada enfoque que no se utiliza en esta sección.

Table 3.2 Comparación de enfoques de aumentación web

Aspecto	Ad-hoc	CS Aug.	CB C&S Aug.	MDWA
Acceso a recursos	Completa	Limitado al navegador	Limitado al navegador y backend independiente	Limitado al navegador y backend independiente
Habilidades de programación	Alta	Alta	Alta	Baja
Nivel de abstracción	Código	Código	Código	Modelos
Reuso	Baja	Baja	Baja	Alta
Productividad	Baja	Baja	Baja	Alta
Administración	Alta	Baja	Baja	Baja
Tecnología del lado del cliente	HTML, CSS and JS frameworks	HTML, CSS and JS frameworks	HTML, CSS and JS frameworks	–
Tecnología del lado del servidor	Web Frameworks (basados en Java, Python, etc)	Web Frameworks (basados en Java, Python, etc)	Web Frameworks (basados en Java, Python, etc)	IFML

4. Discusión técnica

4.1 Introducción

A continuación, se describen todos los aspectos técnicos que fueron resueltos en el desarrollo de la herramienta para la planificación de una producción agrícola. Como se ha mencionado la actividad de planificación consiste en la investigación de diferentes productos alternativos utilizados. Esto conllevará a la navegación de diferentes sitios para recopilar información sobre semillas, pesticidas y fertilizantes, durante esta interacción del usuario el frontend representado a través de un navegador web contará con una extensión que lo asistirá en el proceso de selección de productos. Esta herramienta se basa en una desarrollada para la abstracción y reconocimiento de conceptos WOA [17], a la que se le agrega la capacidad de manejo de dos grandes funcionalidades, por un lado se permite registrar los conceptos identificados con el proceso presentes en las diferentes páginas web en un servidor público al que anteriormente se ha denominado IRAPI. Por otro lado la otra gran modificación es la capacidad de interacción con dicho servidor para lograr la satisfactoria extracción de datos de los elementos utilizando la información de los conceptos registrados por los usuarios o en su defecto el manejo y visualización de los posibles errores ocurridos en el proceso. También se describe un experimento llevado a cabo sobre dos aplicaciones de la herramienta antes mencionada y una encuesta evaluando aspectos de usabilidad y la metodología cotidiana con la que las personas resuelven los problemas planteados en los ejemplos.

4.2 Frontend

La primera limitación encontrada surge con instalación de WOA [14] en modo desarrollo, ya que posee un conjunto de requerimientos que se listan a continuación para instalar y correr el proyecto en el navegador. En este caso el navegador web utilizado es Mozilla Firefox [28] ya que es sobre el que está desarrollado y probado el código fuente de la herramienta.

4.2.1 Requerimientos WOA

La documentación de la herramienta WOA especifica como necesarias el uso de estas herramientas para el desarrollo y carga de la extensión en el navegador.

- NPM ¹
- JPM ²
- Firefox Extension Auto-Installer ³

NPM es el administrador de paquetes para JavaScript y el registro de software más grande del mundo. Este es necesario para instalar JPM, que es una herramienta basada en NodeJS[31] que permite probar, ejecutar y empaquetar add-ons. Firefox addon autoinstaller es una herramienta para los desarrolladores de extensiones que evita el ciclo de prueba habitual: realizar un cambio, crear una nueva compilación de extensiones, instalarla en el navegador (con tiempo de espera y confirmaciones), reiniciar el navegador, probar. Esta extensión recibirá nuevos paquetes de extensión en un puerto TCP, los instalará automáticamente y reiniciará el navegador si es necesario. Esta es el add-on que permite cargar la extensión en el navegador.

¹<https://www.npmjs.com/>

²<https://developer.mozilla.org/fr/docs/Mozilla/Add-ons/SDK/Tools/jpm>

³<https://addons.mozilla.org/es/firefox/addon/autoinstaller/>

4.2.2 Versión de Mozilla Firefox

Uno de los aspectos a tener en cuenta en el desarrollo de extensiones web es la compatibilidad con las versiones del navegador sobre el que se está desarrollando, cada versión se basa en un núcleo de código base que contiene, como es el caso de Firefox[28] el motor central sobre el que corre combinando tecnologías como XUL ⁴, Javascript, XBL ⁵, y C++ ⁶ entre otras.

El 14 de noviembre de 2017 Firefox[28] lanzó una nueva versión llamada Firefox Quantum[29] lo que provocó que la add-on Extension Auto-Installer quede discontinuada al no ser compatible como informa oficialmente en el web store.

Al trabajar en un ambiente de desarrollo y con una herramienta experimental que no es publicada oficialmente en el web store desde donde los usuarios se podrían descargar e instalar el add-on del enfoque libremente, existe una dependencia para poder realizar la carga de la extensión. La utilización esta add-on para la instalación nos genera una limitación en su última versión 1.3 publicada ya que nos acota al uso del navegador sobre el que se puede trabajar a un rango de lanzamientos entre los números 45.0 y 56.*.

La documentación oficial para el desarrollo de extensiones web de Firefox[28] explica como utilizar una herramienta de línea de comandos llamada web-ext ⁷ disponible en el repositorio de NPM. Esta herramienta se basa en la existencia del archivo manifest.json que es un archivo con formato JSON y es el único archivo que debe contener cada extensión que use API de WebExtension según la documentación oficial.

```
{
  "applications": {
    "gecko": {
```

⁴<https://developer.mozilla.org/es/docs/Mozilla/Tech/XUL>

⁵<https://developer.mozilla.org/es/docs/XBL>

⁶C++ es un lenguaje de programación. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. Desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

⁷ <https://www.npmjs.com/package/web-ext>

```
    "id": "pedromslp@gmail.com",
    "strict_min_version": "42.0"
  }
},
"version": "0.1",
"name": "woa"
}
```

4.2.3 Peticiones desde el navegador web

Uno de los puntos principales del enfoque presentado es el soporte backend de las aumentaciones web, para esto se debe utilizar un método que permita el manejo de peticiones disparadas desde el navegador hacia los servicios y la detección del estado de la ejecución, para poder realizar el manejo correspondiente ya sea en caso de éxito o fallo. Basándonos en la característica de que la herramienta sea multiplataforma y corra sobre diversas plataformas, se debió adoptar una solución que siga los lineamientos base de la herramienta WOA sobre el uso de librerías y mecanismos incorporados en el núcleo del código de los navegadores web. Para esto se utilizó la librería client-side estándar JavaScript, para realizar llamados Ajax ⁸, es un término para describir dos capacidades de los navegadores que han estado presentes por años.

Las dos capacidades en cuestión son:

- La posibilidad de hacer peticiones al servidor sin tener que volver a cargar la página.
- La posibilidad de analizar y trabajar con documentos XML.

Para poder realizar esta interacción con los servicios backend se utilizó un objeto JavaScript llamado XMLHttpRequest, que actualmente es un estándar de la W3C⁹. La especificación de este objeto define una API que proporciona funcionalidad de

⁸JavaScript Asíncrono y XML

⁹<https://www.w3.org>

cliente a través de scripting¹⁰ para transferir datos entre un cliente y un servidor. Una característica que posee es que permite realizar las solicitudes síncronas o asíncronas, pero no se recomienda utilizar las síncronas porque, dada la naturaleza inherentemente asíncrona de las redes, hay múltiples formas en que la memoria y eventos se puedan perder usando solicitudes síncronas.

Cuando se ejecuta la herramienta y se realizan las peticiones, éstas tienen como origen el dominio actual sobre el que se encuentra navegando el usuario. Esto quiere decir que todas las solicitudes a los servicios son de dominio cruzado. Esta característica hace que entre en juego una política del modelo de seguridad de las aplicaciones web llamada "política del mismo origen", ésta define que un navegador web permite que los scripts contenidos en una primera página web tengan acceso a los datos en una segunda página web, pero sólo si ambas páginas web tienen el mismo origen. Un origen se define como una combinación de esquema de URI, nombre de host y número de puerto. Esta política evita que un script malicioso en una página obtenga acceso a datos confidenciales en otra página web a través del DOM de esa página.

4.3 Backend

4.3.1 IRAPI

El servicio de IRAPI descrito anteriormente surge ante la necesidad de tener un catálogo público y dinámico de conceptos, en el cual principalmente se puedan registrar y obtener los diversos datos que los usuarios deben registrar para cada servicio que pueda ser cargado. En este caso el servicio de Planificación de Producción lo utiliza para el guardado de la información estructural necesaria para la extracción de los conceptos para cada sitio. Para esto se podría utilizar cualquier framework web pero por facilidad en base a mi experiencia fue desarrollado utilizando el framework

¹⁰Es la capacidad que tiene un lenguaje de programación de soportar scripts

Symfony2 ¹¹, en particular implementa la arquitectura REST¹² por su flexibilidad y aprovecha los cuatro principales principios sobre los cuales se rige su diseño:

1. Utiliza los métodos HTTP de manera explícita
2. No mantiene estado
3. Expone URIs con forma de directorios
4. Transfiere XML, JSON, o ambos

Siguiendo estas pautas el servicio expone públicamente dos métodos que facilitan tanto el guardado como la consulta de los atributos registrados para un dominio en particular, como se define a continuación:

Nombre	Registrar atributo
URL	http://irapi.herokuapp.com/create_concept
Método	POST
Parámetros Data	<pre>{ name : [string], category : [string], tag : [string] url : [string], xpath : [string] }</pre>
Respuesta satisfactoria	Código: 200
Respuesta error	Código: 422

¹¹ <http://symfony.com/>

¹²REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web

Nombre	Obtener atributos
URL	http://irapi.herokuapp.com/get_concepts
Método	GET
Parámetros Url	dominio=[string]
Respuesta satisfactoria	<pre>{ Success : [boolean], Category : [string], Domain : [string], Concepts : [{ CreatedAt: [date], Name: [string], Tag: [string], Xpath: [string] }] }</pre>
Respuesta error	Código: 422

5. Trabajos realizados

5.1 Objetivo

Con el fin de evaluar la viabilidad del enfoque, se llevó a cabo un estudio de usabilidad de las aplicaciones aumentadas con los usuarios finales. El objetivo es analizar los aumentos de la web que se basan en las características del servidor para medir cómo se mejora la usabilidad en contraste con la misma tarea realizada sin el aumento. El estudio se llevó a cabo desde el punto de vista de los investigadores en el contexto de los usuarios finales que diariamente usan aplicaciones web.

Este estudio es preliminar y tiene como objetivo evaluar la usabilidad y el rendimiento de los aumentos generados bajo el enfoque. Por lo tanto, no se consideró evaluar la experiencia del usuario relacionada con el uso de otro enfoque. El lector debe tener en cuenta que el proceso de desarrollo que combina WOA y WebRatio no está bajo estudio.

Durante el estudio, se solicitó a los usuarios que realicen la planificación de la producción de hortalizas por parte de un agricultor presentado en la sección 3.3.3 "Orquestación compleja de conceptos sensibles a la navegación". La planificación requiere investigar la alternativa de diferentes productos como semillas, pesticidas y fertilizantes. La información utilizada en el análisis se recopila de diferentes sitios de comercio electrónico que describen los productos disponibles, incluida su cantidad y precio. Al final, el usuario proporciona un informe que describe los tres productos obligatorios para la producción de hortalizas.

5.2 Sujetos

Los participantes fueron 20 usuarios finales que incluyen profesionales, científicos, contadores y abogados, entre otros. La edad de los participantes se muestra en un diagrama de caja en Figure 5.1. En cuanto a la educación, uno obtuvo un título secundario, 6 de ellos son universitarios y 13 obtuvo un título de licenciatura en sus respectivas profesiones.

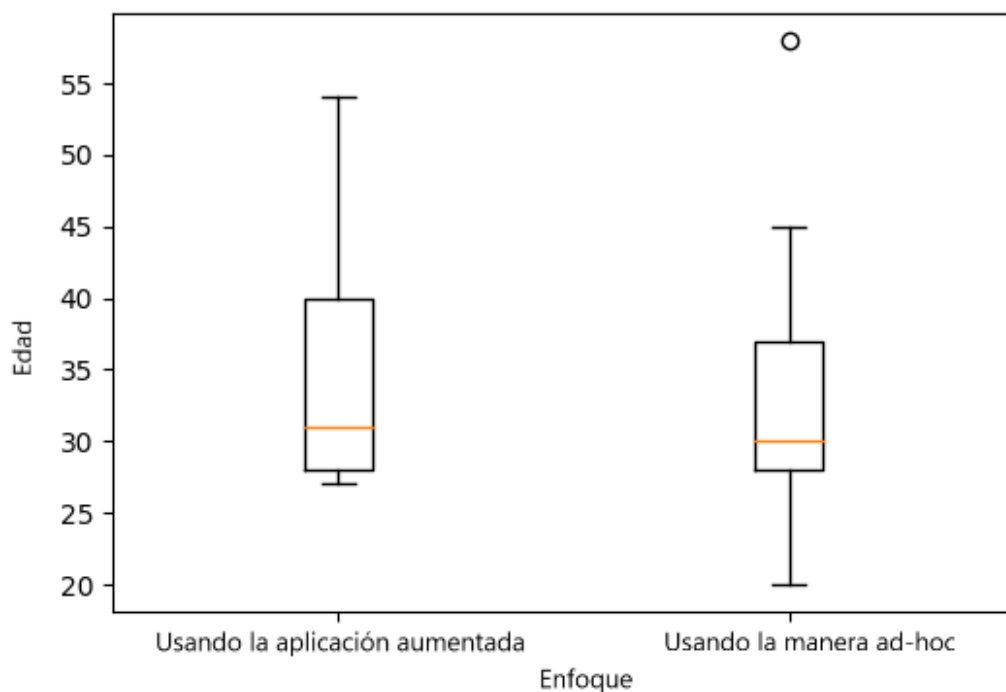


Fig. 5.1 Edad de los sujetos

5.3 Materiales

Las unidades experimentales utilizadas durante el experimento comprenden los sitios web donde se recupera información sobre semillas, pesticidas y fertilizantes, y el aumento de los sitios mencionados mejorados con una característica que ayuda a

planificar la producción de vegetales. El sitio de comercio electrónico es: Arg-Agro ¹, Dinkos ², Easy ³ y AgroPoints ⁴. Además, se pidió que completaran dos cuestionarios: un cuestionario demográfico y un cuestionario SUS.

Se empaquetó el navegador web con la extensión de plugin WOA en una máquina virtual basada en el producto Oracle Virtual Box utilizando una nueva imagen de Microsoft Windows 10. El uso de máquinas virtuales nos permitió garantizar que no hay diferencia en el entorno en el que trabajan los sujetos en. La máquina virtual se suministró con Mozilla Firefox, la extensión del complemento y cualquier script de aumento requerido para el experimento. El entorno virtual se alojó en servidores privados propios y las sesiones de Windows se accedieron de forma remota utilizando TeamViewer ⁵. Las sesiones de los usuarios se grabaron para realizar un análisis profundo sobre la experiencia del usuario de la aplicación a través de la función Microsoft Windows 10, que le permite grabar la pantalla y todas las acciones que se realizan. A partir de la sesión grabada, se extrajo el tiempo consumido por cada participante y se observaron los eventos del mouse y del teclado para un análisis posterior. Para dar soporte a las características del lado del servidor, se utilizó WebRatio Web Platform Community Edition [41]. Como se mencionó anteriormente en la sección 3.3.3, se usó Webratio para modelar la aplicación que permite que el soporte de backend administre la lógica de negocios y persista la información.

5.4 Protocolo

El protocolo de experimento se ejecutó de la misma manera con todos los sujetos. Los sujetos se dividieron en dos grupos utilizando un enfoque (el factor del estudio): el aumento generado con el enfoque o las aplicaciones sin ningún aumento (una forma ad-hoc). En primer lugar, completaron un cuestionario demográfico, luego se les pidió

¹<http://www.arg-agro.com.ar/>, último acceso 12 de febrero de 2018

²<http://www.semillas-organicas.com.ar/>, visitado por última vez el 12 de febrero de 2018

³<https://www.easy.com.ar/webapp/wcs/stores/servlet/es/easyar>, visitado por última vez el 12 de febrero de 2018

⁴<http://www.agropoints.com/>, visitado por última vez el 12 de febrero de 2018

⁵<https://www.teamviewer.com/>, visitado por última vez el 20 de febrero de 2018

que realizaran las tareas de estudio. Se dividieron los sitios web de acuerdo con cada tipo de producto que debería elegirse, los sitios Arg-Agro y Dinkos se proporcionaron para elegir semillas, AgroPoints para fertilizantes y Easy para pesticidas. Las tareas eran simples requisitos que imitaban a un agricultor que planifica su siembra, para esto necesitaban explorar los sitios y seleccionar una semilla de tomate, un fertilizante y un pesticida usando sus propios criterios:

- En los sitios Arg-Agro o Dinko:
 - • Tarea 1 : buscar y elegir una semilla de tomate;
 - • Tarea 2 : explorar la aplicación y agregue el producto a su planificación de producción;
- En el sitio AgroPoint:
 - • Tarea 1 : buscar y elegir un fertilizante;
 - • Tarea 2 : explorar la aplicación y agregar el producto;
 - • Tarea 3 : verifique si la semilla de tomate y el fertilizante seleccionados están disponibles en su informe;
- En el sitio Easy:
 - • Tarea 1 : buscar y elegir un pesticida;
 - • Tarea 2 : explorar la aplicación y agregar el producto;
 - • Tarea 3 : verifique si todos sus productos están disponibles en su informe;

Finalmente, se les hizo algunas preguntas sobre su percepción de la usabilidad de los aumentos generados. Se les proporcionó a través de Google Forms⁶ un cuestionario SUS en línea [5].

⁶https://www.google.com/intl/es-419_ar/forms/about

5.5 Resultados, análisis e implicancia

La evaluación de los cuestionarios SUS [5] correspondiente a la tarea de planificación se muestra usando un diagrama de caja en la figura 5.2. Las puntuaciones de SUS indican que hay una mejora en la satisfacción del usuario cuando se utiliza el complemento sobre la misma tarea, sobre cuando se realiza de forma ad-hoc. Como la distribución de las muestras no es paramétrica, la prueba de Mann-Whitney [25] tiene como resultado un valor p 0,0113 que es menor que el nivel alfa 0,05, y el efecto el tamaño es 0.565 lo que significa un alto significado. Por otro lado, la figura 5.3 muestra un gráfico de diagrama de caja que informa el tiempo dedicado a la planificación de la producción de vegetales. Al estudiar las medidas con la prueba de Mann-Whitney, el valor de p 0.05 (que es igual al nivel alfa) respalda que haya un mejor rendimiento de los usuarios finales que utilizan el enfoque. Los resultados de este trabajo presentan una evidencia preliminar de que un aumento con soporte de servidor introduce una mejora estadísticamente significativa. Sin embargo, un estudio completo debería comprender un experimento controlado donde evaluar el enfoque de desarrollo con un conjunto más amplio de análisis de materia y potencia, y un estudio completo de la experiencia del usuario de los aumentos desarrollados utilizando el enfoque.

5.6 Amenazas a la validez

Existen varias amenazas a la validez que se consideraron durante el diseño del experimento.

Validez de la construcción: el experimento se diseñó para medir cómo el uso de IU aumentadas con un aumento de cliente y servidor mejora la usabilidad de los casos de uso de la aplicación. Con el fin de reducir la complejidad del experimento y la posibilidad de introducción de sesgo, se definió el método (con o sin aumento) como la única variable independiente.

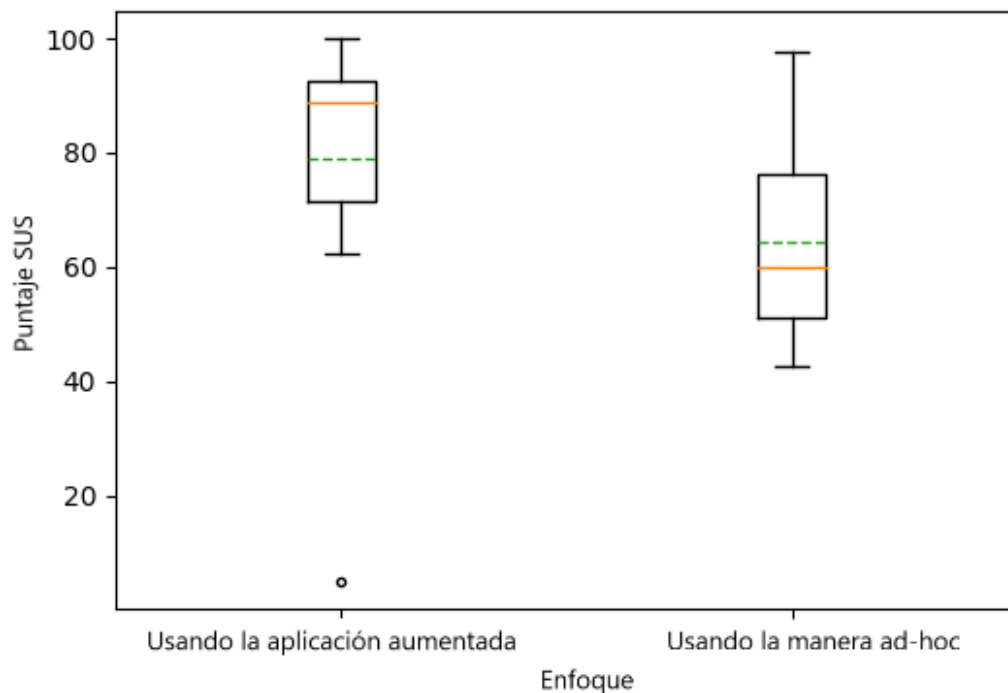


Fig. 5.2 Puntaje SUS

Validez interna: los sujetos fueron seleccionados al azar. El material proporcionado fue el mismo para todos los sujetos. También se verificó que todos los usuarios tuvieran conocimiento básico del uso de la aplicación web. Además, el uso de una máquina virtual para proporcionar un escritorio ayuda a garantizar condiciones homogéneas cuando los sujetos realizan el experimento, así como a evitar el entorno configurado en sistemas operativos y hardware heterogéneos. Sin embargo, podría haber una amenaza en el caso de mala conectividad que perjudica la experiencia de uso de la máquina virtual.

Validez externa: los sujetos fueron usuarios finales mixtos. Un experimento más amplio que considere temas de diferentes culturas que hayan trabajado en dominios comerciales dispares mejorará la generalidad de las afirmaciones.

Conclusión de la validez: el experimento se basó en métricas objetivas evaluadas con todos los datos recopilados sin exclusión para garantizar resultados idempotentes y evitar la pesca de hipótesis. El bajo poder estadístico es una amenaza para este

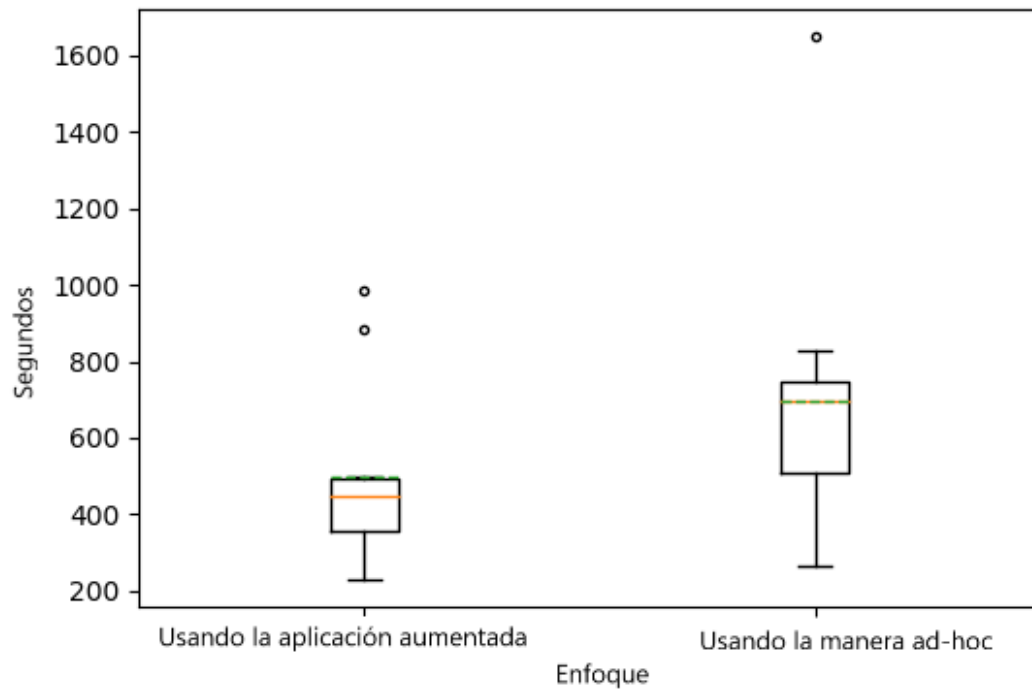


Fig. 5.3 Tiempo dedicado en realizar la tarea

estudio. Por lo tanto, se debe realizar un estudio más amplio con un conjunto más amplio de muestras que ayuden a diluir las irrelevancias del experimento.

6. Trabajos futuros

6.1 Experimento

Como se menciona anteriormente en la sección 5.6, la baja cantidad de muestras analizadas puede resultar en una amenaza a la validez y potencialmente supone irrelevancias en el experimento. Para no caer en un falso positivo, la propuesta para trabajar a futuro es tomar una cantidad mínima de 50 observaciones por condición. Para el protocolo descrito en la sección 5.4 la división en los 2 grupos hace suponer que un conjunto total de 100 sujetos aumentaría las evidencias estadísticas para lograr determinar que el factor de estudio fue exitoso.

6.2 Mejoras en la herramienta

Las mejoras propuestas surgen de observaciones realizadas sobre la experiencia de los usuarios al momento de realizar el experimento. Durante el transcurso de los ejercicios sobre aumento de los sitios con soporte en el servidor, se evidenciaron varios factores potencialmente mejorables. Estos se enumeran a continuación:

- Mejorar la navegabilidad: la tecnología utilizada para efectuar el aumento, en ciertos casos, afectó la navegabilidad cuando los usuarios utilizaron funciones básicas del navegador web, por ejemplo al seleccionar la opción de retroceder una página se retrocedía en el aumento en vez del sitio.
- Ubicación customizable: resulta de interés que la ubicación la aumentación sea configurable para poderse adaptar las necesidades y gustos de cada usuario, por

otro lado, una ubicación fija del aumento puede traer problemas al bloquear visualmente información o funcionalidades del sitio base.

- **Agregar ayudas:** para mejorar la utilización, el entendimiento y el enfoque dado a las herramientas, surge la necesidad de agregar ayudas en la interfaz que puedan ser activadas o desactivadas y que sirvan para guiar al usuario durante su interacción con el sistema.
- **Acceso a la información:** agregando más funciones al servidor, se puede proveer al usuario de acceso a la información histórica que utilizó en sus anteriores sesiones. Para esto se debería implementar un mecanismo de autenticación en la herramienta y así poder guardar toda la información necesaria para luego retribuirle en caso de ser solicitada.

7. Conclusión

En este trabajo se presenta un enfoque novedoso para afrontar el diseño de Aumentación Web con comportamientos del lado del cliente y del lado del servidor. Los aumentos se modelan utilizando IFML, pero el enfoque presentado se puede instanciar con otros enfoques [34] como UWE u OOHDM. El enfoque se basa en la separación de los principios de los conceptos. Por lo tanto, se proporciona el mecanismo de composición para cada modelo (conceptual, navegacional y la interfaz de usuario).

Para describir las posibilidades del enfoque, se describieron algunos ejemplos que respaldan las diferentes necesidades y complejidades de los usuarios en un sitio de comercio electrónico agrícola (es decir, el diseño de una característica de revisión comunitaria y una navegación sensible al concern para planificar la siembra). La implementación de estos ejemplos tuvo como objetivo mejorar las actividades de toma de decisiones. Luego, se presenta una discusión que compara tres enfoques para proporcionar algunas ideas sobre las ventajas y desventajas de producir aumentos: la implementación ad-hoc (requisitos evolutivos), el desarrollo basado en código en ambos, cliente y servidor, y el enfoque dirigido por modelos. La evaluación comprende un estudio preliminar que informa los beneficios del enfoque en la satisfacción del usuario y el rendimiento de los sujetos.

Se planeó y llevó a cabo un experimento de planificación de una producción agrícola, que fue presentado como ejemplo en la sección 3.3.3, para evaluar la percepción de los usuarios finales sobre el proceso de desarrollo utilizando el enfoque. Esto ayuda a mejorar los procesos en términos de eficiencia. Los resultados de este trabajo presentan una evidencia preliminar de que un aumento con soporte de servidor introduce una mejora estadísticamente significativa. Sin embargo, un estudio completo debería

comprender un experimento controlado donde evaluar el enfoque con un conjunto más amplio de sujetos y así analizar la experiencia de los usuarios.

Referencias

- [1] Aoki, Y. and Nakajima, A. (1999). User-Side Web Page Customization. In *Human-Computer Interaction: Ergonomics and User Interfaces, Proceedings of {HCI} International '99 (the 8th International Conference on Human-Computer Interaction), Munich, Germany, August 22-26, 1999, Volume 1*, pages 580–584.
- [2] Asakawa, C. and Takagi, H. (2008). Transcoding. In Harper, S. and Yesilada, Y., editors, *Web Accessibility: A Foundation for Research*, pages 231–260. Springer London, London.
- [3] Bosch, J., Olsson, H. H., Björk, J., and Ljungblad, J. (2013). The Early Stage Software Startup Development Model: A Framework for Operationalizing Lean Principles in Software Startups. *Lean Enterprise Software and Systems*, pages 1–15.
- [4] Brambilla, M. and Fraternali, P. (2014). *Interaction Flow Modeling Language: Model-Driven UI Engineering of Web and Mobile Apps with IFML*. Morgan Kaufmann, Burlington, Massachusetts, USA.
- [5] Brooke, J. (1996). SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- [6] Brusilovsky, P. (2007). Adaptive Navigation Support. In *The Adaptive Web, Methods and Strategies of Web Personalization*, pages 263–290.
- [7] Brusilovsky, P., Kobsa, A., and Nejd, W. (2007). *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*. Springer.
- [8] Ceri, S., Dolog, P., Matera, M., and Nejd, W. (2004). Model-Driven Design of Web Applications with Client-Side Adaptation. In Koch, N., Fraternali, P., and Wirsing, M., editors, *Web Engineering: 4th International Conference, ICWE 2004, Munich, Germany, July 26-30, 2004. Proceedings*, pages 201–214. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [9] Cerny, T., Macik, M., Donahoo, M. J., and Janousek, J. (2015). On distributed concern delivery in user interface design. *Computer Science and Information Systems*, 12(2):655–681.
- [10] DDway (2016). Calculation of the Functional Size and Productivity with the IFPUG method (CPM 4 . 3 . 1). The DDway experience with Webratio. URL http://www.webratio.com/website/documentation/Case_Study_Productivity_with_WebRatio.pdf.
- [11] Díaz, O. and Arellano, C. (2015). The Augmented Web: Rationales, Opportunities, and Challenges on Browser-Side Transcoding. *TWEB*, 9(2):8.

- [12] Díaz, O., Arellano, C., and Iturrioz, J. (2010). Interfaces for Scripting: Making Greasemonkey Scripts Resilient to Website Upgrades. In *Web Engineering, 10th International Conference, {ICWE} 2010, Vienna, Austria, July 5-9, 2010. Proceedings*, pages 233–247.
- [13] Diigo (2017). Diigo. URL <https://www.diigo.com/>.
- [14] Firmenich, D., Firmenich, S., Rivero, J. M., Antonelli, L., and Rossi, G. (2016a). CrowdMock: an approach for defining and evolving web augmentation requirements. *Requirements Engineering*, pages 1–29.
- [15] Firmenich, D., Firmenich, S., Rivero, J. M., Antonelli, L., and Rossi, G. (2016b). CrowdMock: an approach for defining and evolving web augmentation requirements. *Requirements Engineering*, pages 1–29.
- [16] Firmenich, S., Bosetti, G., Rossi, G., and Winckler, M. (2016c). Flexible Distribution of Existing Web Interfaces: An Architecture Involving Developers and End-Users. In *Current Trends in Web Engineering - {ICWE} 2016 International Workshops, DUI, TELERISE, SoWeMine, and Liquid Web, Lugano, Switzerland, June 6-9, 2016, Revised Selected Papers*, pages 200–207.
- [17] Firmenich, S., Bosetti, G. A., Rossi, G., Winckler, M., and Barbieri, T. (2016d). Abstracting and Structuring Web Contents for Supporting Personal Web Experiences. In *Web Engineering - 16th International Conference, {ICWE} 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings*, pages 77–95.
- [18] Firmenich, S., Rossi, G., Urbietta, M., Gordillo, S., Challiol, C., Nanard, J., Nanard, M., and Araujo, J. (2010). Engineering concern-sensitive navigation structures, concepts, tools and examples. *Journal of Web Engineering*, 9(2):157–185.
- [19] Fowler, M., Beck, K., Brant, J., Opdyke, W., and Roberts, D. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, Boston, MA, USA.
- [20] Frajberg, D., Urbietta, M., Rossi, G., and Schwinger, W. (2016). Volatile Functionality in Action: Methods, Techniques and Assessment. In Bozzon, A., Cudre-Maroux, P., and Pautasso, C., editors, *Web Engineering: 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings*, pages 59–76. Springer International Publishing, Cham.
- [21] Gaedke, M. and Gräf, G. (2001). Development and Evolution of Web-Applications Using the WebComposition Process Model. In *Web Engineering*, volume 2016, pages 58–76.
- [22] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [23] Ginzburg, J., Distanto, D., Rossi, G., and Urbietta, M. (2009). Oblivious integration of volatile functionality in web application interfaces. *Journal of Web Engineering*, 8(1):25–47.

- [24] Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00*, pages 241–250.
- [25] Juristo, N. and Moreno, A. M. (2010). *Basics of Software Engineering Experimentation*. Springer Publishing Company, Incorporated, 1st edition.
- [26] Martin, R. C. (2003). *Agile software development: principles, patterns, and practices*. Prentice Hall PTR.
- [27] Mikkonen, T., Systä, K., and Pautasso, C. (2015). Towards Liquid Web Applications. In Cimiano, P., Frasincar, F., Houben, G.-J., and Schwabe, D., editors, *Engineering the Web in the Big Data Era: 15th International Conference, ICWE 2015, Rotterdam, The Netherlands, June 23-26, 2015, Proceedings*, pages 134–143, Cham. Springer International Publishing.
- [28] Mozilla Firefox (2017a). Mozilla Firefox. URL <https://www.mozilla.org/es-ES/firefox/>.
- [29] Mozilla Firefox (2017b). Mozilla Firefox quantum version. URL <https://blog.mozilla.org/blog/2017/11/14/introducing-firefox-quantum/>.
- [30] Niederhausen, M., Sluijs, K., Hidders, J., Leonardi, E., Houben, G.-J., Meißner, K., Van Der Sluijs, K., Hidders, J., Leonardi, E., Houben, G.-J., and Meißner, K. (2009). Harnessing the power of semantics-based, aspect-oriented adaptation for AMACONT. In Gaedke, M., Grossniklaus, M., and Díaz, O., editors, *Web Engineering*, chapter Harnessing, pages 106–120. Springer, Berlin, Heidelberg.
- [31] NodeJS (2017). NodeJS. URL <https://nodejs.org>.
- [32] Popovici, A., Gross, T., and Alonso, G. (2002). Dynamic Weaving for Aspect-oriented Programming. In *Proceedings of the 1st International Conference on Aspect-oriented Software Development, AOSD '02*, pages 141–147, New York, NY, USA. ACM.
- [33] Rossi, G., Nieto, A., Mengoni, L., Lofeudo, N., Silva, L. N., and Distanto, D. (2006). Model-Based Design of Volatile Functionality in Web Applications. In *2006 Fourth Latin American Web Congress*, pages 179–188.
- [34] Rossi, G., Pastor, O., Schwabe, D., and Olsina, L. (2008). *Web Engineering: Modelling and Implementing Web Applications*, volume 12. Springer-Verlag London.
- [35] Torrecilla-Salinas, C., Sedeño, J., Escalona, M., and Mejías, M. (2015). Estimating, planning and managing Agile Web development projects under a value-based perspective. *Information and Software Technology*, 61:124–144.
- [36] Urbietta, M., Firmenich, S., Maglione, P., Rossi, G., and Olivero, M. A. (2017). A Model-driven Approach for Empowering Advance Web Augmentation - From Client-side to Server-side Support. In *APMDWE*. INSTICC, ScitePress.

- [37] Urbietta, M., Oliveira, A., Araújo, J., Rodrigues, A., Moreira, A., Gordillo, S., and Rossi, G. (2014). Web-GIS models: accomplishing modularity with aspects. *Innovations in Systems and Software Engineering*, 10(1):59–75.
- [38] Urbietta, M., Retschitzegger, W., Rossi, G., Schwinger, W., Gordillo, S., and Luna, E. R. (2012a). Modelling adaptations requirements in web workflows. *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services - IIWAS '12*, page 72.
- [39] Urbietta, M., Rossi, G., Distante, D., and Ginzburg, J. (2012b). Modeling, Deploying, and Controlling Volatile Functionalities in Web Applications. *International Journal of Software Engineering and Knowledge Engineering*, 22:129–155.
- [40] Vilain, P., Schwabe, D., and de Souza, C. S. (2000). A Diagrammatic Tool for Representing User Interaction in UML. In Evans, A., Kent, S., and Selic, B., editors, *International Conference on the Unified Modeling Language*, volume 1939 of *Lecture Notes in Computer Science*, pages 133–147. Springer.
- [41] WebRatio (2017). WebRatio Platform. URL <http://www.webratio.com/site/content/en/web-application-development>.
- [42] Wischenbart, M., Firmenich, S., Rossi, G., and Wimmer, M. (2015). Recommender Systems for the People - Enhancing Personalization in Web Augmentation. In *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, IntRS 2015, co-located with {ACM} Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 19, 2015.*, pages 53–60.