

Asistentes virtuales en plataformas 3.0

Javier Medina, Eduardo M. Eisman, Juan Luis Castro

Virtual Solutions & Artificial Intelligence
Parque Tecnológico Ciencias de la Salud. Avenida de la Innovación 1, Edificio BIC, módulo 313. CP 18016,
Granada (España)
{javier.medina,eduardo.eisman,castro}@solucionesvirtuales.net

Resumen: Los asistentes virtuales son herramientas inteligentes que ayudan a los usuarios a buscar información en un conglomerado de recursos web. El despliegue natural de los mismos se realiza en las propias páginas web, donde permiten resolver las dudas de los usuarios formuladas en lenguaje natural usando técnicas de Inteligencia Artificial. En este artículo presentamos las características más relevantes del asistente virtual Elvira y su integración en la página web de la Universidad de Granada. De forma paralela a la aparición de los asistentes virtuales, en la última década, los avances tecnológicos han hecho que el acceso a la información se produzca desde diferentes fuentes, trasladando la necesidad de la asistencia artificial a otros ámbitos. En este trabajo, detallamos la ampliación de los despliegues del asistente virtual Elvira sobre dispositivos móviles y redes sociales.

Palabras clave: asistentes virtuales, redes sociales, dispositivos móviles.

Abstract: Virtual assistants are intelligent tools that help users to find information in a conglomerate of web resources. Their natural deployment is made on the web pages where they allow solving user queries formulated in natural language using Artificial Intelligence techniques. In this paper, we present the most relevant features of the virtual assistant Elvira and their integration into the web page of the University of Granada. Parallel to the development of virtual assistants, in the last decade, technological trends have made the access to information happen from various sources, moving the need of artificial assistance to other areas. In this paper, we detail the extension of the deployments of the virtual assistant Elvira on mobile devices and social networks.

Key words: virtual assistants, social networks, mobile devices.

1. Introducción

La expansión de Internet ha propiciado que una ingente cantidad de información se integre en la web. Estos recursos pueden ser accedidos mediante páginas web que organizan y presentan la información mediante enlaces de navegación. Sin embargo, muchas instituciones o compañías relevantes muestran un elevado conglomerado de datos que es difícil de explorar y buscar.

Por ello, los asistentes virtuales se presentan como eficaces herramientas en la localización de información en las webs. La principal potencialidad

es que los usuarios pueden realizarles consultas en lenguaje natural, al igual que si fueran personas humanas, para navegar y acceder al conglomerado de recursos web [Khentout 2007].

No obstante, es necesario realizar una profundidad interpretación del lenguaje para recuperar el sentido de la pregunta y relacionarlo con una respuesta relevante. Esta interpretación va más allá de lo sintáctico, introduciendo relaciones semánticas entre las entidades referidas en las consultas. En el primer apartado, *Motor de Razonamiento Lingüístico*, describiremos el motor de razonamiento, cuya interpretación incluye las fases de procesamiento del

lenguaje y el diseño de la ontología que modela el conocimiento de cada dominio.

Para que el conocimiento pueda ser recuperado desde diferentes plataformas (web, móviles o redes sociales), el motor de razonamiento se ha encapsulado como un módulo de consulta mediante un servicio web que puede ser accedido desde cualquier dispositivo conectado a Internet. La arquitectura del sistema se detalla en la segunda sección *Arquitectura basada en Servicios Web*.

Los entornos web han representando un entorno propicio para los asistentes virtuales, ya que gran parte del conocimiento de los mismos se encontraban en la red. Sin embargo, el auge de los dispositivos móviles propició la integración de los asistentes virtuales en estos entornos. Respecto a la movilidad, podemos distinguir dos tipos de asistentes: de propósito general y de dominio específico. Los asistentes de propósito general han generado gran expectación gracias al desarrollo de *Siri* [Gruber 2009] por parte de Apple y de *Google Now* [Brad Reed 2012] por parte de Google. Estos tipos de asistentes generales son útiles para conocer el tiempo que va a hacer, cómo llegar a una calle determinada, enviar un SMS de viva voz, etc...

Los asistentes de propósito específicos sirven para resolver las cuestiones sobre un dominio específico, como una tienda, un museo, una institución, etc. En movilidad han sido interesantes en la asistencia a compras [Santangelo 2007] y en la asistencia de personas mayores en entornos de hogar [Ahamed 2007] por su capacidad para resolver problemas concretos in situ. El asistente de la Universidad de Granada de este artículo se enmarcaría en este grupo.

La integración de los asistentes virtuales en redes sociales está tema muy reciente. Actualmente, hemos sido la primera empresa en publicar un asistente virtual integrado en Facebook. Con anterioridad, existían algunos asistentes conversacionales que se integraban en mundos virtuales para dialogar con ellos; tales como [Griol 2010] donde se respondían dudas sobre horarios e información de asignaturas en Second Life.

En la última sección de este artículo, *Despliegue Multiplataforma*, describiremos tres entornos de despliegue para el asistente virtual.

- Web, bajo una interfaz animada que integra emociones.
- Móvil, mediante una conversación natural usando el micrófono y el altavoz del dispositivo.
- social, respondiendo desde las redes sociales a las preguntas de los usuarios.

Todos los ejemplos mostrados en este artículo hacen referencia a Elvira, el asistente virtual de la Universidad de Granada.

2. Motor de Razonamiento Lingüístico

La interpretación del lenguaje es un proceso complejo, que se divide o integra varias fases, tales como la sintaxis o la semántica. Del mismo modo, nuestro asistente virtual incluye dichas etapas en su procesamiento del lenguaje natural.

La primera de ellas es el reconocimiento sintáctico. La principal dificultad de esta etapa, es relacionar sinónimos para que la interpretación sea robusta a términos semejantes. Un claro ejemplo es la estrecha relación de semejanza entre el término “alumno” y “estudiante”. Además, la relación sintáctica puede ir más allá de la sinonimia, involucrando múltiples términos, como “UGR”, “Universidad de Granada” o “nuestra Universidad”.

Nuestro asistente virtual integra un módulo de reemplazos mediante expresiones regulares que establece reglas de acoplamiento de términos, haciendo que aquellos semejantes se identifiquen bajo una única formulación llamada *token*. Este módulo es editable e incluye términos específicos del dominio donde se va a desplegar el asistente (véase la Figura 1).

```
BolognaPlan.GS : (bologna|bologna_plan)
BolognaPlan.GS : (new (((universit(y|ary)|educational) )?
(degree(s)?|plan(s)?|study_plan|qualification(s)?))
BolognaPlan.GS : (european(( higher)? education)? area|
european (space|area)(( for)? higher)? education)?|
EHEA|europe(an)? education)
```

Figura 1. Fragmento del módulo de reemplazos

Una vez unificados los términos del contexto del asistente, es necesario establecer una relación semántica entre ellos, y qué mejor herramienta para definir la semántica, que una ontología *ad hoc* [Noy 2001]. La estructura ontológica que se ha definido tiene como entidad principal los “objetos”. Un objeto tiene varios atributos asociados, tales como nombre, frase de explicación, URL del recurso web, etc. Un ejemplo de objeto sería el “Rector de la Universidad”.

Los objetos, además, incluyen propiedades que detallan sus características y ofrecen una explicación a un nivel de granularidad menor. Por ejemplo, “nombre”, “teléfono”, “correo”, “cargo”, etc. Además, las propiedades también pueden ser otro objeto, de forma que se recupere información lógica de segundo grado a partir de la estructura de la ontología. Por ejemplo, “teléfono del vicesecretario del rector” hace referencia al teléfono de la persona de confianza del rector.

Los objetos, a su vez, disponen de herencia de clases, permitiendo que las instancias hereden propiedades de sus padres. Por ejemplo, la propiedad “institución={Universidad de Granada}” se especifica una única vez para todos los trabajadores de la UGR, y las clases e instancias que la heredan o implementan tienen por defecto esa propiedad y ese valor.

Uno de los aportes más relevantes que incluye nuestra ontología son las recomendaciones y las sugerencias, que permiten establecer relaciones con otros objetos o propiedades. De esta forma, cuando se responde al usuario con la información de un objeto, también se le sugieren una serie de opciones para proseguir con la conversación. Esta relación también permite definir listas de opciones sobre temas generales de información, como “becas”, que hacen referencia a una gran cantidad de objetos.

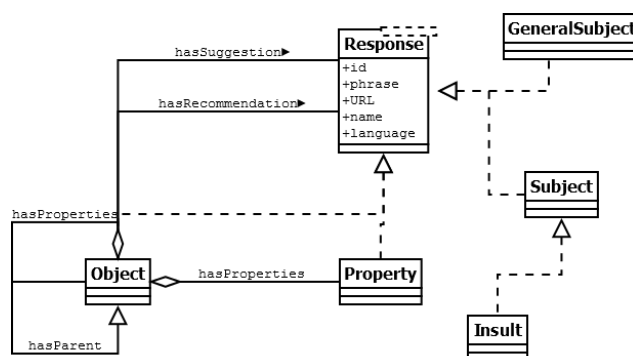


Figura 2. Estructura simplificada de la ontología

El algoritmo encargado de relacionar los *tokens* con los objetos y sus propiedades se define como *Algoritmo de Matching*. La potencialidad de nuestro algoritmo de matching estriba en dar respuesta en base al contexto de la conversación. Para ello, dispone de un módulo de memoria que recuerda los objetos y las propiedades a los que el usuario ha hecho referencia con anterioridad. Por ejemplo, si se pregunta “Rector” y luego “teléfono”, hemos de recuperar la propiedad “teléfono” del objeto anterior “Rector”. La memoria es útil, además, para establecer objetos iniciales de referencia. En nuestro ejemplo, los objetos iniciales son “Universidad de Granada” y “Elvira” (el propio asistente), en ese orden. De esta forma, al iniciar una conversación, si se escribe “teléfono”, el asistente devolverá el número de teléfono de la UGR; mientras que si se escribe “edad” devolverá la edad del asistente.

Para establecer una prioridad entre los diversos objetos de la ontología, hemos definido diferentes clases que establecen un orden de preferencia a la hora de dar una respuesta. Pongamos por ejemplo que un usuario escribe un insulto: “tonta, dime la dirección de la universidad”. En este caso no debemos dar la información que ha solicitado el usuario, si no que debemos recuperar únicamente la respuesta para este insulto leve. Estos temas de primera prioridad se conocen como *temas específicos* y son los más prioritarios: si se disparan, el resto del algoritmo de matching se detiene.

De forma antagónica, es muy útil definir respuestas para conceptos abiertos, ambiguos, o conceptos que agrupan a otros objetos, antes de renunciar a dar una respuesta. Éstos son los *temas generales*, los menos

prioritarios, que usamos para guiar al usuario entre una serie de recomendaciones cuando no se detecta un tema específico, objeto o propiedad (incluso en la memoria). Un ejemplo son “becas”, que no hacen referencia a un objeto concreto como “beca FPI” o “beca Erasmus”, pero que muestra una lista de todas las becas de la Universidad.

El diagrama UML asociado a la ontología puede verse en la Figura 2.

Finalmente, uno de nuestros últimos aportes es que el motor de razonamiento permite integrar varias ontologías, de forma que la búsqueda se realiza iterativamente en cada una de ellas. Esto nos permite especificar diferentes dominios y contextos en los que evaluar las preguntas de los usuarios. En este ejemplo, existe una ontología general que incluye respuestas del asistente a su edad, ocupación, insultos, chistes, etc. Sobre esta ontología situamos otra de dominio específico que resuelve específicamente las cuestiones de la Universidad de Granada. Si quisiéramos añadir un nuevo contexto, por ejemplo, el del Ayuntamiento de la ciudad, sólo tendríamos que integrar una nueva ontología independiente.

A groso modo, el algoritmo de matching puede resumirse en:

1. *Para cada ontología.*
 - a. *Si se reconoce algún tema específico*
 - i. *Devolver respuesta del tema específico.*
2. *Para cada ontología.*
 - a. *Para cada objeto reconocido*
 - i. *Si se reconocen propiedades de ese objeto*
 1. *Devolver las propiedades.*
3. *Para cada ontología.*
 - a. *Para cada objeto anterior de la memoria*
 - i. *Si se reconocen propiedades de ese objeto*
 1. *Devolver las propiedades.*
4. *Para cada ontología.*
 - a. *Si hay objetos reconocidos*
 - i. *Devolver objetos reconocidos.*
5. *Para cada ontología.*
 - a. *Si se reconoce algún tema general*

- i. *Devolver respuesta del tema general.*

6. *Devolver no reconocimiento de pregunta.*

3. Arquitectura basada en Servicios Web

La inmersión tecnológica de nuestros días permite acceder a la información desde cualquier dispositivo y desde múltiples plataformas. Esto exige a los desarrollos una alta escalabilidad y transparencia para que puedan ser desplegados de forma ubicua por los usuarios [Weiser 1991].

En el apartado anterior explicamos el razonamiento interno del asistente para resolver preguntas. Aquí detallaremos la arquitectura y los protocolos de comunicación para que pueda servir a clientes web, móviles y sociales.

Para este fin, el reconecedor ha sido encapsulado en un servicio web [Benslimane 08] que puede ser consultado usando una URL mediante envío de parámetros. El protocolo de comunicación es HTTP y los parámetros de solicitud se especifican mediante el protocolo GET al estilo REST [Richardson 2008].

Estos parámetros representan la consulta del usuario y especifican:

- El texto introducido.
- El idioma.
- Las ontologías de búsqueda.
- El sexo del asistente (para devolver una frase femenina o masculina en función del género).

El motor de razonamiento descrito anteriormente procesa la consulta en base a esos datos y devuelve una estructura de datos en formato JSON [Crockford 2006] con los siguientes atributos:

- Respuesta en modo texto.
- URL del MP3 de la respuesta.
- Identificador del objeto/propiedad reconocida.
- Objetos/propiedades recomendados.
- Estado de las emociones del asistente.

La estructura de parámetros de entrada/salida puede verse en la Figura 3.

Una de las ventajas de la integración mediante servicios web son las sesiones. Éstas dan a conocer la apertura y cierre de la conexión entre las distintas peticiones de los usuarios, lo que nos permite inicializar y añadir objetos o vaciar la memoria respectivamente.

Cada uno de los clientes, que describiremos en las siguientes secciones, se conectará a este único servicio web.

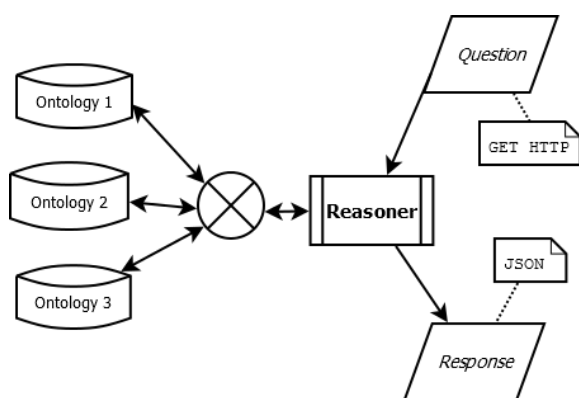


Figura 3. Arquitectura basada en servicios web

4. Despliegue Multiplataforma

La arquitectura mostrada en la sección anterior admite una amplia interconectividad de clientes. A continuación, detallaremos tres clientes: web, móviles y redes sociales que permiten que nos comuniquemos con el asistente virtual desde cualquier plataforma.

4.1. Asistente Virtual Web

Como mucha de la información que integran los asistentes proviene de recursos web, el ámbito innato donde se realizan consultas a los asistentes es sobre las propias páginas web.

La integración en entornos online se realiza sin necesidad de realizar cambios sobre la web, como una capa transparente que se sitúa de forma no invasiva sobre la original.

Esta capa está compuesta por una animación que integra movimientos naturales cuando habla y responde emocionalmente a insultos o halagos.

Cuando un usuario formula su pregunta, las peticiones se envían al servidor web mediante JavaScript usando la tecnología AJAX [Garret 2005]. La respuesta es procesada en el navegador de cada cliente, mostrando el texto asociado a la respuesta, las recomendaciones, etc.

Las recomendaciones se muestran como una lista desplegable a un lado del asistente, de forma que el usuario pueda acceder a ellas en cualquier momento.

Además, gracias a su integración en la web, el asistente puede realizar redirecciones a otras páginas para permitir el acceso a ciertos recursos o mostrar respuestas más detalladas.

Con respecto a la voz del asistente, esta se consigue mediante la reproducción de archivos MP3 previamente generados mediante tecnología *Text To Speech*.

Un ejemplo del asistente virtual web integrado en la página de la Universidad de Granada puede verse en la Figura 4 y a través de la dirección [Elvira Web].



Figura 4. El asistente virtual, Elvira, en la web de la Universidad de Granada.

4.2. Asistente Virtual Móvil

Los dispositivos móviles han revolucionado el acceso a la información gracias a su conectividad con la red global y a su masiva implantación [Hansmann 2003]. En nuestro caso, la integración de asistentes virtuales

en móviles ha permitido el uso de ayuda artificial de forma ubicua.

Nuestra aplicación se ha adaptado a la funcionalidad de las pantallas táctiles y al limitado tamaño de las mismas. En concreto, hemos desarrollado el asistente móvil para la plataforma Android [Android 2011].

Por eso, hemos integrado tres vistas diferentes en la aplicación móvil:

- *Central*. Que incluye a su vez:
 - o la funcionalidad principal para escribir o dictar la pregunta
 - o la presentación en modo texto de la respuesta
 - o las sugerencias
- *Izquierda*. Una lista de las recomendaciones de la respuesta.
- *Derecha*. Un navegador web integrado que permite visualizar y navegar por los recursos web que muestra el asistente virtual.

El cambio entre las vistas se realiza mediante un simple desplazamiento en la pantalla táctil.

En los dispositivos móviles, gracias a la integración de la tecnología *Automatic Speech Recognition*, el usuario puede realizar sus consultas de forma oral. Para ello, hemos incluido el botón de un micrófono que captura la voz y la transforma en texto. De forma análoga, la respuesta es reproducida como audio por los altavoces del dispositivo usando tecnología *Text To Speech*.

Como hemos detallado con anterioridad en el Apartado 3, la integración de los diferentes clientes se realiza mediante servicios web. En el caso de los dispositivos móviles, se realizan peticiones mediante HTTP y GET manteniendo una misma sesión en cada uso de la aplicación. Esta sesión permite que el razonador haga uso de la memoria durante la conversión con el usuario.

Los datos de la respuesta, en formato JSON, son procesados por el dispositivo móvil y son presentados en las tres vistas descritas anteriormente (véase Figura 5).



Figura 5. El asistente virtual Elvira, en dispositivos Android.

El asistente virtual Elvira en Android puede consultarse e instalarse desde la siguiente dirección [Elvira Android].

4.3. Asistente Virtual Social

Las redes sociales permiten la distribución de información entre personas en base a sus relaciones personales. El auge de las mismas hace que los usuarios pasen cada vez más tiempo inmersos en ellas, y Facebook se ha convertido en la página más visitada desplazando a Google. Las redes sociales se han convertido en una nueva forma de acceso a la información [Viswanath 2009].

La integración de los asistente virtuales dentro de las redes sociales es algo totalmente novedoso, y Virtual Solutions ha sido la primera empresa en integrarlo. En concreto, se ha realizado una integración dentro de la red social Facebook [Facebook 2006].

La principal dificultad del asistente social ha radicado en el cambio de paradigma respecto a los asistentes móviles o web.

En las redes sociales no existen las mismas conversaciones que en la web o en el móvil. En lugar de conversaciones etéreas, las conversaciones sociales son permanentes en el tiempo y quedan registradas por la red.

En función del grado de visibilidad, varios usuarios pueden participar en ellas, convirtiéndose en un asistente multiconversacional. Por ejemplo, un usuario podría escribir “quién es el rector” y el

asistente respondería “El Rector es Francisco González Lodeiro”. Otro usuario diferente podría continuar la conversación preguntando “cuál es su función”, a lo que el asistente respondería, en referencia al Rector detectado con anterioridad, “El Rector es la máxima autoridad académica...”.

Tampoco existe una sesión que identifique el inicio y el fin de la conversación. Sin embargo, hemos realizado una analogía entre las conversaciones web o móviles gracias a los *posts*. Cada uno de los *posts* de nuestro asistente en Facebook se convierte en una conversación independiente donde se aplica una memoria individual en cada uno de ellos.

Además, Facebook no transmite en tiempo real los cambios en las conversaciones. Aunque esta red social dispone de un canal de cambios en tiempo real, aún no integra los comentarios en los *posts*, y el retardo del canal es de varios segundos. Para solventar este problema, realizamos lecturas periódicas cada pocos segundos para saber si han llegado nuevos comentarios.

Finalmente, para poder acceder y publicar en redes sociales, es necesario hacer uso de una aplicación *ad hoc* que monitorice una página de Facebook [Graph API 2012]. Aunque las aplicaciones se usan de forma genérica por cualquier usuario, la nuestra es una aplicación para un único usuario “virtual” que simula mirar continuamente su muro y responder a las preguntas que le sugieren. Este usuario virtual, es nuestro *Asistente Virtual Social*, y puede monitorizar varias páginas Facebook simultáneamente, siempre y cuando sea administrador.

Así, a groso modo, el *Asistente Virtual Social* puede verse como un adicto usuario que cada pocos segundos comprueba si hay mensajes o *posts* nuevos en su muro. Como estas consultas se realizan de forma continuada, se han optimizado para recuperar y tratar sólo nuevos mensajes mediante el lenguaje de consulta *Facebook Query Language* (FQL).

En el caso de que los haya, se consulta al razonador teniendo en cuenta los comentarios previos, y se escribe la respuesta en el *post* indicado (véase Figura 6). La resolución de preguntas se realiza internamente mediante la comunicación con servicios web que hemos descrito en la sección 3.

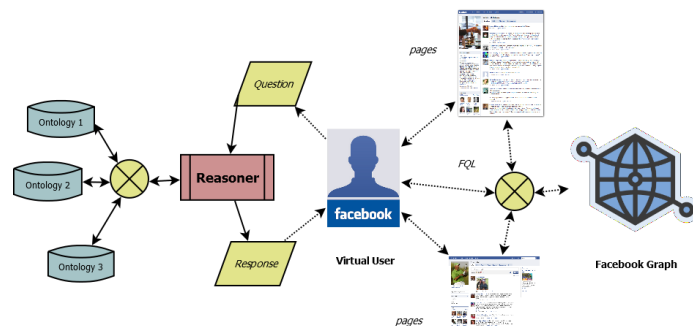


Figura 6. Arquitectura del Asistente Virtual Social.



Figura 7. El asistente virtual Elvira, en Facebook.

En el caso de Facebook, los recursos web que sirven de respuesta se integran dentro del mensaje mediante enlaces cortos. La comprensión se ha realiza en tiempo de ejecución con el servicio de Bitly [Bitly 09].

El asistente virtual Elvira en Facebook puede verse en la Figura 7 y consultarse en la siguiente dirección [Elvira Facebook].

5. Conclusiones

La inmersión tecnológica de nuestros días permite acceder a la información desde cualquier dispositivo y múltiples plataformas. Esto exige a los desarrollos una alta escalabilidad y transparencia para que

puedan ser desplegados de forma ubicua por los usuarios. En este artículo se ha detallado la construcción del asistente virtual de la Universidad de Granada, Elvira, que ha sido desplegado con éxito en plataformas móviles, web y sociales.

En primer lugar, se han descrito las técnicas basadas en Inteligencia Artificial para construir un *Motor de Razonamiento Lingüístico*, entre las que destacan el estudio de la sinonimia, la interpretación semántica mediante ontologías y el uso de memorias de contexto.

Para que el *Motor de Razonamiento Lingüístico* pueda ser consultado desde las diferentes plataformas, se ha integrado un servicio web que facilita su consulta de forma remota mediante protocolos estándar como JSON y HTTP.

Finalmente, se ha validado su correcto funcionamiento en las diferentes plataformas usando distintos formatos de presentación y consulta.

En entornos web, con una imagen animada que responde a emociones otorgando una apariencia con alto nivel de realismo. Además, el asistente es totalmente inocuo situándose de forma transparente sobre los recursos web.

Los dispositivos móviles han permitido introducir la tecnología ASR, logrando una comunicación totalmente verbal sin interferencia de un teclado. Los recursos web se muestran en un navegador integrado que puede mostrarse y ocultarse a demanda.

Finalmente, el asistente virtual Elvira se ha integrado dentro de la red social Facebook, respondiendo en tiempo real a las preguntas que le plantean en su muro. Estas conversaciones persistentes posibilitan que varios usuarios puedan intervenir en una conversación.

Además, el asistente virtual Elvira se ha desplegado de forma efectiva en estos entornos y sirve de ayuda a los alumnos de la Universidad de Granada día a día. Un ejemplo de una conversación de real entre un alumno y nuestro asistente es la siguiente:

Usuario: Hola

Elvira: Hola. Estoy aquí para proporcionarle la información que necesite sobre la Universidad de Granada. ¿En qué le puedo ayudar?

Usuario: Quieres sexo?

Elvira: ¡Vaya cosas que dice! Le pediría que no fuese tan desagradable, recuerde que soy un ente virtual y estoy aquí sólo para comentar cosas que hagan referencia a la Universidad de Granada

Usuario: Que es ets?

Elvira (redireccionando a la web de las ETSI): ETSI es la abreviatura de Escuela Técnica Superior de Ingeniería. A continuación, por si buscaba alguna información sobre ellas en concreto, le muestro un listado de todas las ETSI de la Universidad de Granada.

Usuario: Asignaturas que se dan en arquitectura?

Elvira (redireccionando a la web de Arquitectura): Esta página contiene la información sobre las asignaturas y el profesorado del Grado en Arquitectura. Si pulsa sobre el nombre de alguna asignatura podrá ver los datos de la misma y el profesorado que la imparte, incluyendo sus correos electrónicos, horarios de tutoría...

- *Plan de estudios de arquitectura*
- *Información sobre arquitectura*
- *Centro de arquitectura*
- *Salidas de arquitectura*

Usuario: gracias

Elvira: De nada, mi obligación es resolver todo tipo de dudas sobre la UGR.

- *Estudios de la UGR*
- *Servicios de la UGR*
- *Actividades de la UGR*

Referencias

[Android 2011] Developers, A. (2011). What is android?. vol. 2. 2011.

- [Ahamed 2007] Ahamed, Sheikh I., et al. "Wellness assistant: a virtual wellness assistant using pervasive computing." Proceedings of the 2007 ACM symposium on Applied computing. ACM, 2007.
- [Benslimane 2008] Benslimane, D.; Dustdar, S.; Sheth, A.. "Services Mashups: The New Generation of Web Applications". IEEE Internet Computing 10 (5): 13-15. doi:10.1109/MIC.2008.110. (2008).
- [Bitly 2009] Bitly. www.bitly.com . [Cassell 2000] Cassell, J. (2000). Embodied conversational agents. MIT press.
- [Brad Reed 2012] Google Now wins 'Innovation of the Year', runs laps around 'outdated' Siri. BGR. 2012-11-15. Retrieved 2012-12-26.
- [Crockford 2006] D. Crockford. The application/json Media Type for JavaScript Object Notation (JSON). 2006.
- [Elvira Android] <https://play.google.com/store/apps/details?id=es.solucionesvirtuales.android>
- [Elvira Facebook] <https://www.facebook.com/ElviraAsistenteVirtualUniversidadDeGranada?fref=ts>
- [Elvira Web] <http://tueris.ugr.es/elvira/>
- [Facebook 06] Facebook. www.facebook.com .
- [Garrett 2005] Garrett, Jesse James. "Ajax: A new approach to web applications." (2005).
- [Graph API 2012] GRAPH, A. P. I. Facebook developers. URL: <https://developers.facebook.com/docs/reference/api/>. Marzo, 2012.
- [Griol 2010] Griol, D., et al. "A Conversational Academic Assistant for the Interaction in Virtual Worlds." Distributed Computing and Artificial Intelligence. Springer Berlin Heidelberg, 2010. 283-290.
- [Gruber 2009] Gruber, T. R. "Siri, a Virtual Personal Assistant. Bringing Intelligence to the Interface." (2009).
- [Hansmann 2003] Hansmann, U. (2003). Pervasive computing: The mobile world. Springer-Verlag New York Incorporated.
- [Khentout 2007] Khentout, C., Djoudi, M., & Doudi, L. (2007). Roundup of Graphical Navigation Helpers on the Web.
- [Noy 2001] Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Ferguson, R. W., & Musen, M. A. (2001). Creating semantic web contents with protege-2000. Intelligent Systems, IEEE, 16(2), 60-71.
- [Richardson 2008] Richardson, Leonard, and Sam Ruby. RESTful web services. O'Reilly Media, 2008.
- [Viswanath 2009] Viswanath, B., Mislove, A., Cha, M., & Gummadi, K. P. (2009, August). On the evolution of user interaction in facebook. In Proceedings of the 2nd ACM workshop on Online social networks (pp. 37-42). ACM.
- [Weiser 1991] Weiser, M. (1991). The computer for the 21st century. Scientific american, 265(3), 94-104.