

# VENTURE: Hacia un entorno para el entrenamiento del Desarrollo Global de Software

Miguel J. Monasor, Aurora Vizcaíno, Mario Piattini

Grupo de Investigación Alarcos,  
Instituto de Tecnologías y Sistemas de Información,  
Escuela Superior de Informática,  
Universidad de Castilla-La Mancha,  
13071, Ciudad Real, España  
[MiguelJ.Monasor@gmail.com](mailto:MiguelJ.Monasor@gmail.com)  
[{Aurora.Vizcaino, Mario.Piattini}@uclm.es](mailto:{Aurora.Vizcaino, Mario.Piattini}@uclm.es)

**Resumen.** La necesidad reciente de las universidades por proporcionar una adecuada formación, tanto práctica como teórica, en Desarrollo Global de Software (DGS), ha llevado al desarrollo de nuevos métodos de formación. Muchas de las propuestas existentes tratan de involucrar a estudiantes de diferentes culturas en el desarrollo de proyectos conjuntos. Sin embargo, llevar a cabo estas prácticas es complejo, puesto que requieren un alto grado de coordinación entre universidades.

En este trabajo se propone una arquitectura que evita estos problemas a través de la simulación. Dicha arquitectura hace uso de Agentes Virtuales que simulan ser miembros de un proyecto multicultural y permiten a los estudiantes enfrentarse a problemas realistas, con la ventaja de que pueden practicar en cualquier momento. En este trabajo se describe la arquitectura VENTURE, que consta de diferentes módulos para tratar problemas culturales y de lenguaje. Finalmente se describe la forma de evaluación prevista.

**Palabras clave:** Desarrollo Global de Software; Desarrollo Distribuido de Software; Educación; Entorno de Aprendizaje; Simulador

## 1 Introducción

Actualmente existen numerosas propuestas en la literatura tanto de universidades como del ámbito empresarial orientadas a impartir docencia en Desarrollo Global de Software (DGS) [1]. Las técnicas frecuentemente empleadas consisten en imitar las condiciones de trabajo reales coordinando a alumnos de universidades de diferentes países en el desarrollo conjunto de un software [2]. De esta forma, los participantes afrontan problemas típicos de comunicaciones, colaboración y trabajo en equipo típicos de estos entornos [3] y que se ven agravados por las diferencias culturales y lingüísticas [4].

Sin embargo, estos métodos conllevan problemas: limitación del número de participantes, dificultad para formar grupos apropiados, dificultad para establecer relaciones con otras instituciones, ineficiencia de las comunicaciones, desmotivación

de los alumnos [5], o la falta de rigor para entrenar problemas que durante interacciones reales suelen aparecer de forma aleatoria [2]. Dados estos inconvenientes, nuestra investigación se ha centrado en proponer una arquitectura que facilite el desarrollo de simuladores para el proceso de enseñanza/ aprendizaje de DGS.

En este trabajo se presenta VENTURE; un entorno que facilita la impartición de cursos teóricos y prácticos basados en la simulación de escenarios de DGS. A través de estas simulaciones, los participantes adquieren habilidades comunicativas y de trabajo en equipo sin necesidad de interactuar con participantes reales. La interacción se lleva a cabo mediante Agentes Virtuales (AVs) que simulan ser de diferentes culturas y juegan un rol en el escenario de entrenamiento comunicándose con el estudiante a través de chat o email. Una de las principales ventajas del uso de AVs radica en que los usuarios pueden utilizar la aplicación en cualquier momento sin depender de la disponibilidad de compañeros de otras culturas. Además, se dispone de una arquitectura flexible que permite simular un amplio abanico de situaciones predefinidas que no requieren la colaboración de distintas organizaciones. Otra ventaja es que los escenarios de entrenamiento son repetibles y todos los estudiantes pueden tener experiencias similares, lo cual no es posible en entornos reales, puesto que cada estudiante debe asumir un rol concreto. Por otro lado, al trabajar con AVs en lugar de con otros estudiantes, éstos se sienten más libres de probar alternativas y descubrir por su propia iniciativa.

Por otra parte, VENTURE utiliza un Agente Compañero (AC), que es un tipo especial de AV que guía la interacción y que corrige a los usuarios durante la simulación, como si de un alumno brillante se tratara. De esta forma se evita que el alumno pueda quedar “atascado” en un escenario sin poder avanzar. Además, de ser una fuente de retroalimentación para el alumno.

## 2 Arquitectura de VENTURE

En esta sección se explican los principales componentes de VENTURE, que sigue una arquitectura cliente servidor, detallada en la Fig. 1, donde la *aplicación e-learning* constituye el núcleo del servidor y consta de los siguientes componentes:

- **Repositorio de recursos** (1): Contiene todo el material asignado al alumno; es decir, tanto las lecciones teóricas como las simulaciones que podrá llevar a cabo.
- **Área de tareas** (2): para planificar y gestionar las actividades prácticas asignadas a los alumnos. También podrán realizar envíos de entregables según los plazos establecidos y que serán evaluados automáticamente.
- **Módulos foro y wiki** (3, 4): Se utilizan para interactuar con los instructores o con otros estudiantes del mismo curso y así intercambiar impresiones o conocimientos.
- **Área de evaluación** (5): Para realizar exámenes y cuestionarios que permitan al alumno conocer sus progresos y autoevaluarse.
- **Módulo pedagógico** (6): contiene las diferentes estrategias empleadas para entrenar habilidades de acuerdo a las necesidades específicas de cada estudiante.
- **Problemas Culturales**: La base de datos de *Problemas Culturales* (7) contiene las estructuras textuales que pueden emplearse en los simuladores para entrenar problemas culturales y diferencias que pueden afectar a la comunicación en DGS. Dichas estructuras se almacenan en lenguaje VTRML (*VenTuRe Markup*

*Language*), en el que se definen todos los elementos necesarios para la interacción, incluyendo el conocimiento conversacional de los AVs. Concretamente se incluye información sobre el tipo de problema que se pretende entrenar, la dificultad y las acciones que el AC debe realizar para corregir al alumno.

De este modo, se consideran aquellos problemas culturales presentes en DGS tales como: el efecto silencio (tendencia a esconder o distorsionar información negativa tornándola positiva), el efecto sordo (reticencia a escuchar malas noticias), el uso de estilo directo frente al indirecto, etc. Estos problemas pueden aparecer en mayor o menor medida en función las culturas presentes en la interacción. En este sentido, en [6-7] se estudian las dimensiones culturales que establecen el grado en el que dichas culturas se diferencian. Éste último contempla las siguientes dimensiones: asertividad, colectivismo institucional, colectivismo de grupo, distancia al poder, igualdad de género, orientación al desempeño, aversión al riesgo orientación al futuro y orientación humana. Dependiendo de las costumbres culturales del alumno y teniendo en cuenta la cultura que simulan los AVs, VENTURE enfocará las simulaciones a entrenar aquellas situaciones problemáticas que pueden ocurrir durante una interacción entre ambas culturas.

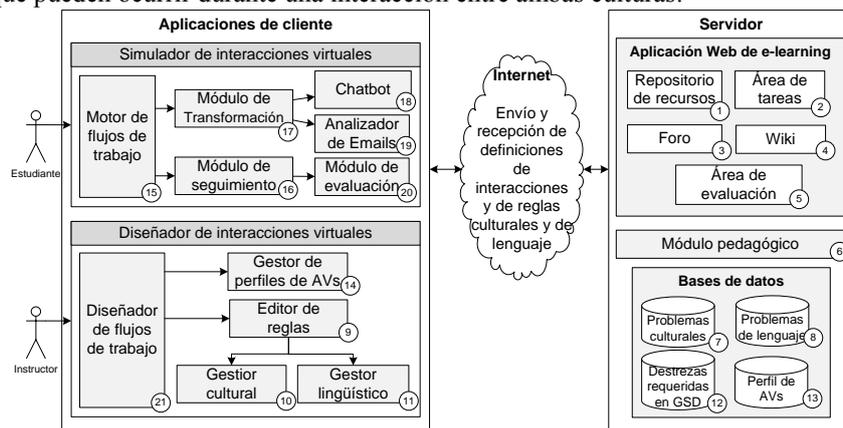


Fig. 1. Arquitectura de VENTURE.

- Problemas de Lenguaje:** La base de datos de *Problemas de Lenguaje* (8) contiene los problemas lingüísticos que pueden aparecer cuando los participantes interactúan textualmente en un lenguaje no nativo. Al igual que en el caso anterior, estas estructuras se almacenan en formato VTRML y pueden utilizarse en la definición de cualquier simulador. Así, en el caso de un alumno español interactuando en inglés, consideramos errores comunes como [8]: el incorrecto uso de *false friends*, ausencia de terceras personas, mal uso o ausencia de formas pasivas, uso incorrecto de formas verbales, sobreutilización de determinados verbos genéricos (*do, have, make, put, take*, etc.). La definición de los problemas culturales y de lenguaje se lleva a cabo a través del *Editor de Reglas* (9) (en la aplicación cliente), a través de los módulos de *gestión cultural* (10) y *gestión lingüística* (11).
- Módulo de Destrezas Requeridas en DGS:** Las *destrezas requeridas en DGS* se almacenan igualmente en base de datos (12) como estructuras VTRML. Este

conocimiento se clasifica en base a las siguientes destrezas que los alumnos deben desarrollar en DGS, de acuerdo a nuestra revisión sistemática de la literatura [1]: comunicación informal, capacidad de improvisación, conocimiento del lenguaje y diferencias culturales, liderazgo, resolución de conflictos, gestión del tiempo, habilidades colaborativas, destrezas para ganar la confianza del interlocutor, capacidad de negociación y comunicación escrita, gestión de la ambigüedad y capacidad para evaluar la información de forma crítica.

- **Perfil de AVs (13)**: contiene información sobre la apariencia, emociones y gestos de los AVs y VCs. El objetivo de los AVs es enseñar a los alumnos a entender y reaccionar apropiadamente ante las costumbres y gestos de otras culturas. Esta información se gestiona a través del *gestor de perfiles de AVs* (14).
- **Motor de Flujo de Trabajo (15)**: es responsable de ejecutar los escenarios de entrenamiento en la parte de cliente. Interpreta el contenido de las definiciones VTRML y extrae el conocimiento conversacional así como las reglas culturales y de lenguaje definidas. Este proceso de extracción de la información se lleva a cabo a través del *Módulo de transformación* (17), de forma que la información resultante sea interpretable por el *Sistema de chatbot* (18), en el caso de las interacciones síncronas, y por el *Analizador de emails* (19), en el caso de las interacciones asíncronas. El *Módulo de seguimiento* (16) se encarga de almacenar las acciones de los usuarios para su posterior análisis.
- **Módulo de Evaluación (20)**: se encarga de recoger información sobre las acciones y progresos del alumno con el simulador para determinar las destrezas en las que dicho alumno necesita un mayor entrenamiento y poder adecuar su entrenamiento.
- **Diseñador de Flujos de Trabajo (21)**: permite la definición visual de las interacciones virtuales sin necesidad de tener conocimiento del lenguaje VTRML. De esta forma, las interacciones siguen un flujo secuencial compuesto por un conjunto de fases que contienen los detalles específicos de la conversación para el contexto de dicha fase.

### 3 Interacciones Virtuales

En VENTURE, las interacciones se pueden llevar a cabo a través de chat o email:

**Interacciones Síncronas**: están limitadas en tiempo y el objetivo del alumno consiste en obtener la mayor cantidad de información posible durante la duración de la interacción minimizando los errores culturales y lingüísticos cometidos. En la Fig. 2 se muestra una interacción en la que un estudiante español, jugando el rol de analista, chatea con un cliente virtual de Rusia. De acuerdo a las dimensiones de House, los españoles tienen un alto grado de asertividad en comparación con los rusos, por lo que el simulador se centra en enseñar a los usuarios a ser menos directos en su forma de expresarse y tratar de decir las cosas de forma que suene más educada y menos agresiva para el receptor.

**Interacciones Asíncronas**: En este caso las interacciones virtuales se definen como una secuencia de emails que se van a intercambiar durante la simulación. La estructura de cada email se define igualmente en formato VTRML. Los emails enviados por el alumno se analizan verificando su estructura y organización

comprobando la correcta interacción tanto lingüística como cultural. Este análisis se lleva a cabo utilizando patrones lingüísticos y palabras clave, de momento no se utilizará análisis del lenguaje natural, este sería uno de los posibles trabajos futuros.

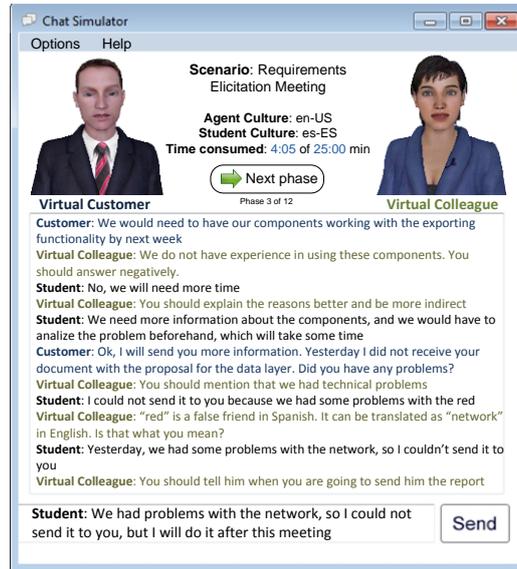


Fig. 2. Ejemplo de conversación a través del Chat.

## 4 Conclusiones

En este trabajo se ha presentado una arquitectura orientada al desarrollo de las habilidades necesarias en DGS a través de la simulación de distintos escenarios. La principal ventaja de VENTURE consiste en que los estudiantes son independientes y no dependen de otros compañeros o instituciones para llevar a cabo el entrenamiento. Además las simulaciones están controladas y rigurosamente definidas y orientadas a entrenar destrezas específicas, lo que no se consigue fácilmente en entornos reales, permitiendo al usuario jugar diferentes roles en el proceso de desarrollo. Otra ventaja es que los usuarios reciben correcciones y sugerencias en el mismo momento en el que cometen los errores y pueden repetir las simulaciones varias veces.

Una ventaja adicional es que usando VENTURE no se pone en riesgo ningún proyecto real y permite simular situaciones difíciles y conflictivas que en otras circunstancias no sería posible reproducir.

## 5 Plan de Trabajo

Como parte del trabajo futuro pretendemos recoger evidencias del uso de VENTURE en varias universidades y multinacionales que permitan demostrar su eficacia. Por

tanto, actualmente estamos en el proceso de preparación de las simulaciones adecuadas, así como de las encuestas y entrevistas que puedan servir para mejorar el entorno. En el futuro nos centraremos en responder a las siguientes preguntas de investigación:

- ¿Entienden los alumnos los objetivos de los entrenamientos?
- ¿Consideran los alumnos que han mejorado sus destrezas en DGS?, ¿Desarrollan habilidades comunicativas y adquieren sintaxis y vocabulario?
- ¿Consiguen alcanzar los objetivos docentes?, ¿Son adecuadas las restricciones temporales establecidas para las simulaciones?
- ¿Se sienten los alumnos motivados interactuando con AVs?, ¿Qué problemas ocurren durante la interacción con AVs?
- ¿Consideran los alumnos que la arquitectura es realmente útil?, ¿Cuál es la percepción de los estudiantes sobre la usabilidad de VENTURE?

Además, durante el proceso de evaluación se realizará un caso de estudio comparando el rendimiento de estudiantes que hayan usado la arquitectura con estudiantes que no lo hayan hecho.

**Agradecimientos.** Este trabajo está financiado por el proyecto PEGASO/MAGO (Ministerio de Ciencia e Innovación MICINN y Fondos FEDER, TIN2009-13718-C02-01). También por los proyectos MEVALHE (HITO-09-126) y ENGLOBAS (PII2I09-0147-8235), financiados por la Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), y Fondos FEDER, así como por GLOBALIA (PEII11-0291-5274) (Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha) y ORIGIN (IDI-2010043 (1-5)) financiado por CDTI y FEDER.

## Referencias

1. Monasor, M.J., A. Vizcaíno, M. Piattini, and I. Caballero. *Preparing students and engineers for Global Software Development: A Systematic Review*. in *International Conference on Global Software Development (ICGSE 2010)*. 2010. Princeton, NJ, USA.
2. Daniels, M., et al. *RUNESTONE, an International Student Collaboration Project*. in *IEEE Frontiers in Education Conference*. 1998. Tempe, Arizona: IEEE
3. Monasor, M.J., M. Piattini, and A. Vizcaíno, *Challenges and Improvements in Distributed Software Development: A Systematic Review*. *Advances in Soft. Engineering*, 2009. p. 14.
4. Abufardeh, S. and K. Magel, *The impact of global software cultural and linguistic aspects on Global Software Development process (GSD): Issues and challenges*, in *4th International conference on New Trends in Information Science and Service Science (NISS)*. 2010: Gyeongju, South Korea. p. 133 – 138.
5. Soller, A., *Supporting Social Interaction in an Intelligent Collaborative Learning System*. *International Journal of Artificial Intelligence in Education (IJAIED)*, 2001. **12**: p. 40-62.
6. Hofstede, G. and G.J. Hofstede, *Cultures and organizations: software of the mind*. 2nd ed, ed. McGraw-Hill. 2005, New York, NY, USA
7. House, R.J., et al., *Culture, Leadership, and Organizations: The GLOBE Study of 62 Societies*. 2004: Sage Publications. 848
8. Shachaf, P., *Cultural diversity and information and communication technology impacts on global virtual teams: An exploratory study*. *Inf. Manage.*, 2008. **45**(2): p. 131-142.