

PROPUESTA DE PROTOCOLO DE INTERPRETACIÓN GRÁFICA PARA EL USO DE DIAGRAMAS DE DISEÑO DE SISTEMAS POR DISMINUIDOS VISUALES Y NO VIDENTES

Hernán Amatriain, Nicolás Perez, Santiago Bianco, Federico Ribeiro, Sebastian Martins, Darío Rodríguez, Hernán Merlino

Grupo de Investigación en Sistemas de Información. Departamento de Desarrollo Productivo y Tecnológico. Universidad Nacional de Lanús. <http://sistemas.unla.edu.ar/sistemas/gisi/>

hamatriain@gmail.com, santiago.bianco.sb@gmail.com, smartins089@gmail.com, darodriguez@unla.edu.ar

Introducción

La actual Ley Nacional de Educación Superior hace responsable al Estado del acceso a la educación universitaria a todo aquel que quiera hacerlo y cuente con la formación y capacidad requerida, y garantizar la accesibilidad al medio físico, servicios de interpretación y los apoyos técnicos necesarios y suficientes, para las personas con discapacidad [Ley 25573, 2002]. También confiere a las instituciones universitarias la autonomía académica e institucional para formular y desarrollar planes de estudio, de investigación científica y de extensión y servicios a la comunidad incluyendo la enseñanza de la ética profesional y la formación y capacitación sobre la problemática de la discapacidad.

Sin embargo los desarrollos en algunas áreas de menor demanda aún son escasos, como es el caso de los disminuidos visuales y no videntes. En particular, el estudio de las carreras de Sistemas de información / Informática, exige la utilización de modelos gráficos para el diseño de funcionalidades y datos. La utilización de estos modelos se hace extremadamente difícil para el estudiante no vidente, y muy complicada la tarea del educando para que el

estudiante los comprenda y utilice para su posterior evaluación, según lo estipula las estructuras curriculares de estas carreras.

Existe un caso puntual en la Universidad Nacional de Lanús, un estudiante no vidente, cursante de la carrera Licenciatura en Sistemas del Departamento de Desarrollo Productivo y Tecnológico. Éste alumno, en su adolescencia perdió la vista y actualmente es difícil para él la comprensión de gráficos de sistemas.

No existe un procedimiento para evaluar y entrenar a estos estudiantes en el manejo de formalismos gráficos, presentándose siempre como un desafío al docente que enfrenta tal situación.

El presente trabajo busca desarrollar un protocolo de comprensión gráfica para facilitar la tarea del docente en explicar estos modelos y la del estudiante no vidente comprenderlos y utilizarlos en la resolución de problemas, para su posterior evaluación.

Estado del arte

El 30 de abril de 2002 se publica en el Boletín Oficial la Ley N° 25573, modificatoria de la

Ley 24521, Ley de Educación Superior (Boletín Oficial N° 29888). Entre las modificaciones que introduce, está la que hace referencia a la “responsabilidad indelegable” del Estado de la prestación del servicio de educación superior de carácter público, que debe garantizar la accesibilidad al medio físico, servicios de interpretación y los apoyos técnicos necesarios y suficientes para las personas con discapacidad [Ley 25573, 2002]. Otras modificaciones que incorpora son las de asegurar el derecho de estudiantes con discapacidad de acceder a los servicios de interpretación y los apoyos técnicos necesarios y suficientes durante las evaluaciones, y la de “formar y capacitar científicos, profesionales, docentes y técnicos, capaces de actuar con solidez profesional, responsabilidad, espíritu crítico y reflexivo, mentalidad creadora, sentido ético y sensibilidad social, atendiendo a las demandas individuales, en particular de las personas con discapacidad”.

En la Universidad Nacional de Lanús, se viene haciendo un trabajo de interpretación y apoyo técnico a personas con discapacidad desde el Programa de Inclusión Universitaria para Personas con Discapacidad de la Dirección de Bienestar Universitario dependiente de la Secretaría de Cooperación y Servicio Público. Esta iniciativa se fundamenta en la necesidad de abordar una problemática social más amplia, que remite a las oportunidades educativas que poseen aquellos sectores de la población históricamente excluidos del Sistema de Educación Superior, partiendo de una nueva manera de concebir, diseñar e implementar políticas universitarias, asentada en el reconocimiento, respeto y afirmación de las diferencias inherentes a la población estudiantil,

congruente con la política de democratización del conocimiento que lleva adelante la Universidad Nacional de Lanús. Hasta hace poco tiempo la discapacidad era pensada como una cuestión que concernía exclusivamente a las personas afectadas, quienes con asistencia y protección podían rehabilitarse a fin de adaptarse al medio y lograr la integración social. Centrado en los aspectos médicos, en lo patológico, este modelo médico-céntrico de la discapacidad perdía de vista los derechos de este grupo heterogéneo, como personas y como ciudadanos, ignorando que lo que está en juego es la calidad de vida y la construcción de una sociedad realmente inclusiva, capaz de enriquecerse de la diversidad.

Sin embargo se ha presentado un caso que pone en evidencia la falta de madurez que se tiene en ciertas áreas específicas con discapacidades puntuales. Particularmente se presentó una prueba difícil al equipo interdisciplinario cuando el estudiante no vidente de la carrera de Licenciatura en Sistemas del Departamento de Desarrollo Productivo y Tecnológico, tuvo que aprender y ser evaluado en algunas técnicas y metodologías de modelado de sistemas que son esencialmente gráficas como los Diagramas de Flujo Nassi-Shneiderman, Diagramas Entidad Relación (DER), Lenguaje Unificado de Modelado (UML), que son solo algunas de las herramientas gráficas que forman parte curricular de la mencionada carrera (acreditada por Resolución CONEAU 1089/12). Efectivamente, no existen herramientas ni procedimientos especiales para evaluar a un no vidente en estas áreas específicas.

No existe un procedimiento para evaluar y entrenar a estos estudiantes en el manejo de

formalismos gráficos, presentándose siempre como un desafío al docente que enfrenta tal situación. Se han buscado distintas formas de darle una solución a este y otros inconvenientes que se presentan a los alumnos con estas características, trabajando con equipos interdisciplinarios, pero más allá de muy buenas ideas y apoyo que ha recibido el estudiante en las distintas áreas, el diseño utilizando modelado con gráficos ha sido hasta el momento un escollo insoslayable.

En un intento de solucionar este problema instalado en la actualidad, se buscaron herramientas especiales, encontrando que algunas posibles soluciones como ser el uso de [PSeInt, Planttext, PlantUML] son parciales e insuficientes. Las herramientas que existen en la actualidad y que pueden utilizarse de apoyo al diseño de modelos gráficos de sistemas (datos y funciones, u objetos y diagramas UML).

El inconveniente con las herramientas mencionadas es que son incompletas en cuanto por un lado pueden generar gráficos a partir de un código, pero no pueden realizar el trabajo inverso, haciendo imposible que un ciego interprete un diagrama; y por otro lado, no abarcan todos los diagramas de modelado de sistemas, dejando afuera, por ejemplo, los Diagramas Entidad Relación (DER) que se utilizan para la modelización de datos (para la posterior creación de la Base de Datos).

En este contexto es que el presente trabajo busca diseñar un proceso de enseñanza y evaluación de no videntes para que puedan comunicar ideas a través de diagramas de modelado de sistemas por medio de un

intérprete y desarrollar las herramientas software necesarias para darle al estudiante mayor independencia y al docente más agilidad para la enseñanza y evaluación.

Solución propuesta

El presente trabajo busca desarrollar un protocolo de comprensión gráfica y un conjunto completo de herramientas para que un no vidente pueda interactuar con un grupo de estudio o docente (en un ambiente de aprendizaje o evaluación) o un grupo de desarrollo (en un ambiente laboral) a través de estos diagramas de modelado de sistemas.

Objetivos específicos

Objetivo Especifico 1: Desarrollar un lenguaje de interpretación gráfica que pueda codificar los diagramas de modelado de sistemas existentes más utilizados de forma tal que pueda ser interpretado por un no vidente, comprendiendo un gráfico así especificado o que pueda escribir el código que genera un diagrama específico.

Objetivo Especifico 2: Desarrollar un proceso de evaluación basado en el lenguaje de interpretación gráfica para que los docentes cuenten con un protocolo o metodología de actuación a la hora de evaluar a un disminuido visual en el área de utilización de diagramas de modelado de sistemas.

Protocolo de interpretación gráfica

El desarrollo de un protocolo para interpretación gráfica de diagramas de diseño

de sistemas tiene dos objetivos. Por un lado, se busca conseguir un lenguaje a través del cual un no vidente pueda comprender un diagrama de diseño de sistemas (DFD, DER, UML, etc) que haya sido codificado con el mismo; y por otro lado ser la base para la creación de un conjunto de herramientas que puedan pasar tanto del lenguaje a un gráfico y viceversa.

En el presente apartado se presenta el protocolo que implementa el logro del primer objetivo. La idea, básicamente, consiste en que para cada diagrama de sistema exista una codificación específica. De esta manera, a partir de una porción de código puede obtenerse el gráfico correspondiente. Al aplicar el protocolo de interpretación gráfica puede conseguirse la comunicación entre un docente entrenado y un estudiante disminuido visual o no vidente. Así, por ejemplo, el docente le asigna al estudiante un problema para que resuelva a través de un Diagrama de Entidad-Relación (DER), y el alumno (no vidente) desarrolla la solución a través de este lenguaje. Posteriormente, el docente utiliza el protocolo para pasar el código obtenido por el estudiante al DER y evaluarlo.

La codificación de un diagrama de sistema a través del presente protocolo, debe tener en cuenta el objetivo de un gráfico, y la construcción de conocimiento de un individuo que carece de visión. Debe tenerse en cuenta que la representación interna del conocimiento de un no vidente es muy distinta a quien pudiendo ver, analiza las relaciones existentes en un diagrama (por ejemplo un DER) con una simple inspección visual. Los gráficos como son los diagramas de clases, DER, casos de uso, DFD, etc., además de ser el formalismo de un diseño, buscan que con un “golpe de vista” el

diseñador tenga una primera impresión de la parte de sistema que se está modelando, o la pueda comunicar. Además de ello, un diseño con formalismo gráfico constituye una fuente de consulta permanente, de tal modo que el diseñador rápidamente puede observar alguna relación o característica que no terminó de memorizar o haya olvidado. Por ello es un desafío conseguir que un no vidente pueda interpretar un formalismo gráfico a partir de un código y que el mismo sea una fuente de consulta rápida. La forma de percibir de un no vidente un código escrito es de manera “lineal” a partir de un audio y al cual se accede secuencialmente. La solución que se ha planteado a este inconveniente, es la creación de un índice de rápido acceso, de tal forma que el individuo disminuido visualmente pueda navegarlo y detenerse en el elemento que desee revisar (sobre el cual quiere “refrescar la memoria”).

Los diagramas sobre los cuales se ha desarrollado el protocolo hasta la fecha son los siguientes:

- Diagrama de Flujo de Datos (DFD): formalismo de diseño funcional estructurado que utiliza tres herramientas:
 - Diagrama de Contexto (DC): establece la frontera del sistema, delimitando las responsabilidades del diseñador y cuáles son las entidades que se comunican con el sistema.
 - Tabla de Eventos (TE): un evento es aquel suceso que activa o estimula una respuesta en el sistema. Una tabla no es fácil de imaginar para un no vidente. Conceptualmente es más simple representar la tabla como un conjunto de elementos, donde cada elemento es una

fila de la tabla (un evento) y cada columna una característica de este elemento.

- Diagrama de Sistemas (DS): representa el detalle de las funcionalidades a través de procesos que implementan las respuestas a los eventos.
- Diagrama de Entidad-Relación (DER): es la primera aproximación al diseño de una Base de Datos relacional.
- Diagrama de Clases: se utiliza UML.
- Diagrama de Casos de Uso: se utiliza UML.
- Diagrama de Secuencia: se utiliza UML.

Diagrama de Contexto:

La forma de codificar el diagrama de contexto es indicando los flujos por sus nombres y de dónde hacia dónde se dirigen, anteponiendo el elemento generador (Entidad Externa o Sistema) y poniendo posteriormente el elemento receptor. Así, por ejemplo, si la entidad externa Cliente realiza una solicitud al sistema de generación de turnos, el flujo se indica como:

Cliente: Solicitud: Sistema de Turnos;

Donde se cumple con la sintaxis:

<Entidad Externa>: <Flujo>: <Sistema>;

Cada línea de flujo finaliza con punto y coma para delimitarla. La sintaxis e índice son las siguientes:

Diagrama de Contexto:

<Entidad Externa>: <Flujo>: <Sistema>;

<Sistema>: <Flujo>: <Entidad Externa>;

Index:

<Entidad Externa>

Nota aclaratoria: el índice tendrá una lista de las entidades externas, de manera que al seleccionar cualquiera de ellas (el acceso es

más rápido que buscar secuencialmente en el código completo) se tiene acceso a los flujos que parten de ella al sistema o viceversa.

Tabla de Eventos:

Se describe cada evento como un elemento u objeto con una serie de características.

Tabla de Eventos:

Evento <Nro>:

Tipo: <Externo / Temporal>

Entidad Externa: <Nombre EE / vacío>

Descripción: <Breve descripción del evento>

Estímulo: <Flujo que activa al sistema / vacío>

Respuesta: 1. <Respuesta 1>, 2. <Respuesta 2>

Función asociada: <Nombre función asociada>;

Index:

Evento <Nro> (<Descripción>)

Diagrama de Sistemas:

El elemento que debe describirse aquí son los procesos que representan la funcionalidad que implementa el sistema para atender a los eventos definidos en la Tabla de Eventos.

Diagrama de Sistemas:

Proceso <Nro>: //Nota: el número de proceso se corresponde unívocamente con el número de evento.

Nombre: <Nombre del proceso> //Nota: corresponde a la función asociada del evento.

Evento Asociado: (<temporal / externo>) <Descripción del evento>

Entidad Externa: <Nombre de todas las EE asociadas, se desprende de la descripción>

Demoras: <Nombre de las demoras asociadas, se desprende de la descripción>

Descripción: <Breve descripción coloquial de la funcionalidad que debe implementar el sistema, de manera secuencial>

Flujos:

1. <Entidad Externa>: <Flujo>: <Proceso>,
2. <Demora>: <Flujo>: <Proceso>,
3. <Proceso>: <Flujo>: <Entidad Externa>;

Index:

<Nombre Proceso>

Subindex (por cada elemento del índice):

<Descripción>

<Flujos>

<Acceso completo>

Diagrama Entidad-Relación:

El DER se representa a través de las definiciones de todas las entidades primero (con sus respectivos atributos), y posteriormente se definen las relaciones entre entidades. Se indexan solamente las entidades con un subíndice que indique si el usuario de esa entidad quiere conocer sus atributos o las relaciones con otras entidades.

Diagrama Entidad-Relación:

Entidades:

Entidad <Nombre>:

<Clave / Obligatorio / Opcional>: <Nombre Atributo>;

Relaciones:

Entidad <Nombre A>: <Relación>: Entidad <Nombre B>

Entidad <Nombre B>: <Relación>: Entidad <Nombre A>;

Nota: la relación debe ir acompañada por la modalidad y cardinalidad (número mínimo y máximo de relaciones entre entidades) debe indicarse la relación en ambos sentidos por cada par de entidades relacionadas.

Diagrama de Clases:

Los diagramas de clases tienen cinco secciones. La primera donde se definen todas las clases

con nombre y tipo (si es una clase abstracta o una interface, por ejemplo). La segunda sección tendrá la lista de atributos. También se detalla: acceso (si es privado, público, protegido o estático), el tipo (tipo de variable) y el nombre. La tercera sección tendrá los métodos de clase. Estas tres secciones se repiten para cada clase del sistema. Una vez definidas todas las clases en ese orden (clase, atributos, métodos nuevamente clase, atributos y métodos), comienza la cuarta sección: relaciones donde se especifica el tipo de relación entre dos clases (herencia, agregación, etc.) y la cardinalidad (uno a muchos, etc.). Una vez terminadas las relaciones se define el índice, que puede ser por clase o relaciones.

Diagrama de Clases:

Clase <Nro>:

Clase (<clase/clase abstracta/interface >):

<nombre clase>;

Atributos:

<acceso>:<tipo>:<nombre>;

Métodos:

<acceso>:<nombre>(<tipo argumento: argumento 1, tipo argumento: argumento 2>):<tipo retorno>;

Nota: en acceso se especifica si el atributo o método es público, privado, protegido, estático o abstracto (éste último para métodos)

Clase <Nro>:

Clase (<clase/clase abstracta/interface >):

<nombre clase>;

Atributos:

<acceso>:<tipo>:<nombre>

<acceso>:<tipo>:<nombre>;

Métodos:

<acceso>:<nombre>(<tipo: argumento 1, tipo: argumento 2>):<tipo retorno>;

Relaciones:

< ClaseA>: <cardinalidad>: <relación>:

<cardinalidad>: <Clase B>;

Index:

<Nombre Clase>

<Relaciones>

Nota: relación puede ser herencia, composición, agregación

Diagrama de Casos de Uso:

Los casos de uso se especifican comenzando por el actor, e indicando la relación que tiene con otros casos de uso.

Diagrama de Casos de Uso:

<Actor>: < Nombre Caso de Uso>

<Nombre Caso de Uso>: <Include / Extends>:

< Nombre Caso de Uso>;

Index:

<Actor>

< Nombre Caso de Uso>

Nota: el diagrama de casos de uso debe complementarse con la descripción del caso de uso.

Diagramas de Secuencia:

Este diagrama describe cada una de las secuencias del sistema. Se identifica cada una de manera numérica e indica el caso de uso asociado. Luego, el desarrollo de la misma se describe como una sucesión de flujos finalizados con punto y coma.

Diagrama de Secuencia:

Secuencia <Nro>:

Caso de Uso: <Caso de Uso asociado>

Desarrollo:

<Actor>: <Tipo mensaje> <Mensaje>

(<Argumentos>): <Nombre Objeto Receptor>

<Nombre Objeto Emisor>: <Tipo mensaje>

<Mensaje>(<Argumentos>): <Nombre Objeto Receptor>;

Index:

<Casos de Uso asociados>

A continuación se observan algunos ejemplos de los primeros casos. En la figura 1 se aprecia un ejemplo genérico de un diagrama de contexto.

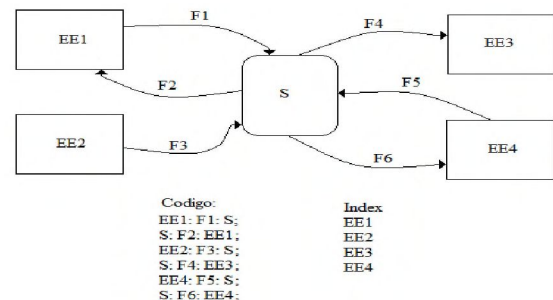


Figura 1: Ejemplo de Diagrama de Contexto

La figura 2 muestra un diagrama de sistemas.

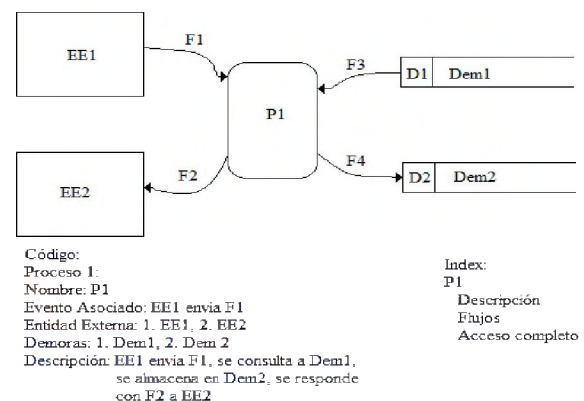


Figura 2: Ejemplo de Diagrama de Sistema

Experiencia

Si bien se está trabajando con el estudiante ciego para la implementación de los protocolos

de interpretación gráfica, aún no se probaron todos los desarrollados, y la instrucción de cada protocolo que se le va enseñando implica una devolución del estudiante que enriquece al mismo.

Actualmente está cursando la materia Ingeniería de Software I y ha demostrado poder resolver problemas con el protocolo para Diagramas de Flujo de Datos, aún no se continuó con el resto de los protocolos debido a los tiempos mismos del proceso de enseñanza.

Ha podido resolver satisfactoriamente problemas de complejidad cada vez mayor, diseñando Diagramas de Contexto, Tablas de Eventos y diagramas de Sistemas. Si bien el avance es a un ritmo menor que el resto de sus pares videntes, los avances son claros y la resolución de problemas bien concreta.

Para el final del cuatrimestre, se espera haber podido trabajar en los protocolos para DFD, DER, Diagramas de Clases, Casos de Uso y Diagramas de Secuencia, además de poder probar una herramienta para implementar Diagrama de Clases que ya fuera desarrollada.

Conclusiones

Se ha desarrollado un protocolo para codificar los diagramas de modelado de sistemas para los formalismos de DFD (DC, TE y DS), DER, Diagrama de Clases, Casos de Uso y Diagrama de Secuencia.

Se han probado los primeros protocolos (DC, TE y DS) con resultados positivos. A medida que los protocolos son probados se van modificando para mejorar la experiencia del

estudiante no vidente, siendo éste de un gran valor para depurar el lenguaje de interpretación gráfica.

Se continuará con la implementación y prueba del resto de los diagramas cuyos protocolos ya fueron desarrollados a lo largo del presente año. Deben desarrollarse más protocolos que continuarán probándose en los siguientes años.

Finalmente se desarrollarán las herramientas informáticas (aplicaciones, servicios web) que implementen estos protocolos para automatizar la generación de gráficos a través de código, y viceversa. Ya hay desarrollada una aplicación para Diagrama de Clases que aún debe ser probada, y se espera desarrollar las herramientas que implementen los protocolos de DER y DFD este año, y probarlos, ir generando el resto de las herramientas en los años subsiguientes.

Referencias

- ANSI/IEEE, (2007). *Draft IEEE Standard for software and system test documentation*. ANSI/IEEE Std P829-2007.
- Argimón J. (2004). *Métodos de Investigación Clínica y Epidemiológica*. Elsevier España, S.A. ISBN 9788481747096.
- Arnett, K. (2003). *Teacher adaptations in core French: A case study of a one grade 9 class*. The Canadian Modern Language Review, 60 (2), 173-198
- Basili, V. (1993). *The Experimental Paradigm in Software Engineering*. En Experimental Software Engineering Issues: Critical Assessment and Future Directions (Ed. Rombach, H., Basili, V., Selby, R.). Lecture Notes in Computer

- Science, Vol. 706. ISBN 978-3-540-57092-9.
- Beardon, C; Lumsden, D; Holmes, G (1991). Natural Language and Computational Linguistics. An introduction. Ellis Horwood. Berge, N. & Berge, Z. (1998). *Integration of Disabled Students into Regular Classrooms in the United States and in Victoria, Australia*. The Exceptional Child, 35 (2), 107-117.
- Booth, T. y Ainscow, M. (2000). *Índice de inclusión, desarrollando el aprendizaje y la participación en las escuelas*. Bristol: CSIE UNESCO.
- Casanova, M. y Cabra, M. (Coords.) (2009). *Educación y personas con discapacidad: presente y futuro*. Madrid: Fundación ONCE.
- Constitución Nacional (1994). *Constitución de la Nación Argentina. Art. 75: Declaraciones, Convenciones, y Pactos complementarios de derechos y garantías*.
- Correa, P. (2010, diciembre). *Los mapas táctiles y diseño para todos los sentidos*. Trilogía. Ciencia, Tecnología, Sociedad, (22), 77-87.
- Delrieux, C. (2003). *Introducción a la Computación Gráfica*. Departamento de Ingeniería Eléctrica, Universidad Nacional del Sur, Agosto, 2003.
- Echeita, G. y Duk, C. (2008). *Inclusión educativa*. Revista Electrónica Iberoamericana sobre Calidad, Eficacia y Cambios en Educación, 6 (2). Recuperado de <http://redalyc.uaemex.mx/pdf/551/55160201.pdf>
- Edler, R. (2008, enero-junio). *Políticas de la educación especial*. Revista Intercontinental de Psicología y Educación, 10 (001), 15-28.
- Edward, A. (2000) *Interactive Computer Graphics: a Top-Down Approach With OpenGL* 2nd edition, AddisonWesley, 2000.
- Fernández Fernández, G.; Sáez Vacas, F. (1995). *Fundamentos de informática: [lógica, autómatas, algoritmos y lenguajes]*. Editorial Anaya Multimedia.
- Foley, J.D.; van Dam, A.; Feiner, S.K.; Hughes, J.F. (1995). *Computer Graphics: Principles and Practice*, 2nd edition, AddisonWesley 1995
- García Martínez, R., Britos, P. (2004). *Ingeniería de Sistemas Expertos*. Editorial Nueva Librería. ISBN 987-1104-15-4.
- Giné, C. (2001). *Inclusión y sistema educativo*. Presentado en el III Congreso "La atención a la diversidad en el sistema educativo", Universidad de Salamanca-Instituto Universitario de Integración en la Comunidad (INICO).
- Grishman, R. (1991). *Introducción a la Lingüística Computacional*. Visor.
- González López, P. y García-Consuegrea Bleda, J. (1998). *Informática Gráfica*. Ediciones de la Universidad de Castilla-La Mancha-Cuenca, 1998.
- González, M. (2008). *Diversidad e inclusión educativa: algunas reflexiones sobre el liderazgo en el centro escolar*. Revista Electrónica Iberoamericana sobre Calidad, Eficacia y Cambios en Educación, 6 (2) <http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=55160208>
- Hernández, C. (2011). *Desarrollo de las concepciones educativas de las personas*

- con discapacidad visual*. La Habana: Pueblo y Educación.
- IEEE, (1997). *IEEE Standard for Developing Software Life Cycle Processes. IEEE Std 1074-1997* (Revisión of IEEE Std 1074-1995; Replaces IEEE Std 1074.1-1995)
- Jacobson, I., Ng, P. W., McMaha, P. E., & Jaramillo, C. M. Z. (2013). *La esencia de la ingeniería de software: El núcleo de Semat*. Revista Latinoamericana de Ingeniería de Software, 1(3), 71-78.
- Ley de Educación Superior (2002). Ley Nacional Nro. 25573 (modifica Ley 24521). Publicada en el Boletín Oficial del 30-abr-2002, Número: 29888, Página: 1
- May, H. & Bridger, K. (2010). *Developping and embedding inclusive policy and practice in higher education*. York, Reino Unido: Higher Education Academy.
- McLean, P., Heagney, M. & Gardner, K. (2003). Going Global: The implications for students with a disability. Higher Education Research & Development, 22 (2), 217-228.
- Moreno Sandoval, A. (1998). *Linguística Computacional*. Síntesis.
- Oktaba, H., Garcia, F., Piattini, M., Ruiz, F., Pino, F., Alquicira, C. (2007). *Software Process Improvement: The Competisoft Project*. IEEE Computer, 40(10): 21-28. ISSN 0018-9162.
- ONU (2006). *Convención sobre el derecho de las personas con discapacidad*. Sede de las Naciones Unidas en Nueva York, Diciembre 2006.
- OpenGL: sitio oficial disponible en <https://www.opengl.org>.
- Orlansky, M. (1982). *Education of Visually Impaired Children in the USA*. Current Issues in Service Delivery, the Exceptional Child, 29 (1), 13-20.
- Planttext: para diagramas UML. Sitio oficial disponible en <https://www.planttext.com>.
- Plantuml: para diagramas UML. Sitio oficial disponible en <http://plantuml.com>.
- PSeInt: para diagramas Nassi-Schneiderman. Sitio oficial disponible en <http://pseint.sourceforge.net>
- Quijano, G. (2008). *La inclusión: un reto para el sistema educativo costarricense*. Revista Educación, 32 (1), 139-155.
- Riveros, H. y Rosas, L. (1985). *El Método Científico Aplicado a las Ciencias Experimentales*. Editorial Trillas. México. ISBN 96-8243-893-4.
- Rumbaugh, J., Jacobson, I., Booch, G. (1999). *The Unified Modeling Language, Reference Manual*. Addison Wesley, ISBN-10: 02-0130-998-X.
- Sabato J, Mackenzie M. (1982). *La Producción de Tecnología: Autónoma o Transnacional*. Instituto Latinoamericano de Estudios Transnacionales - Technology & Engineering. ISBN 9789684293489.
- Shieber, S.M. (1989). *Introducción a los Formalismos Gramaticales de Unificación*. Teide.
- Stainback, S. y Stainback, W. (1999). *Aulas inclusivas*. Madrid: Narcea, SA de Ediciones.
- UNESCO (2009). *Informe de seguimiento de la EPT en el mundo 2009: Superar la desigualdad por qué es importante la gobernanza*. París.