

Motivación: EndoDiag, EndoDiag II y EndoDiagMAS

Sistemas de Apoyo al Aprendizaje Diagnóstico en Endodoncia:

Departamento de Sistemas e Informática

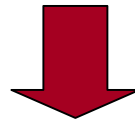
Facultad de Ciencias Exactas, Ingeniería y Agrimensura -UNR

Cátedra de Endodoncia

Facultad de Odontología – UNR

Proyecto EndoDiag:

- Importancia del diagnóstico en endodoncia
- Necesidad de mejorar y jerarquizar la formación de grado y postgrado



EndoDiag como herramienta en el proceso de enseñanza-aprendizaje

EndoDiag:

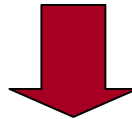
Sistema basado en conocimiento de apoyo al diagnóstico de la patología pulpar y periapical.

Principales características:

- Sistema modularizado.
- Chequeo de la información ingresada.
- Coherencia entre el diagnóstico emitido por el usuario y el alcanzado por el sistema.
- Implementado en Kappa-PC

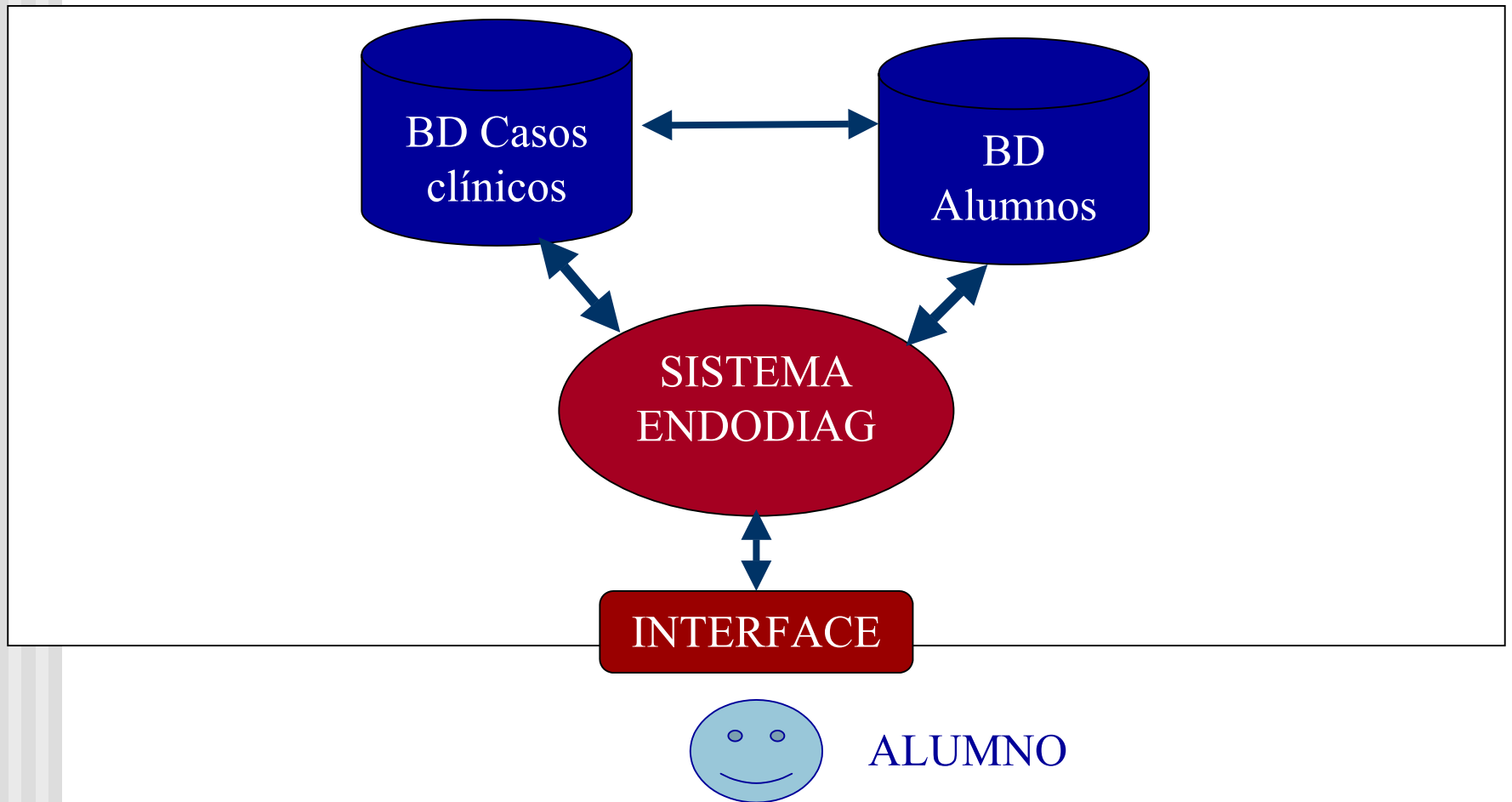
EndoDiag II

- Base de casos clínicos extraídos de la práctica profesional.
- Perfiles de usuario: permite adaptar al sistema eligiendo el caso más adecuado para un alumno

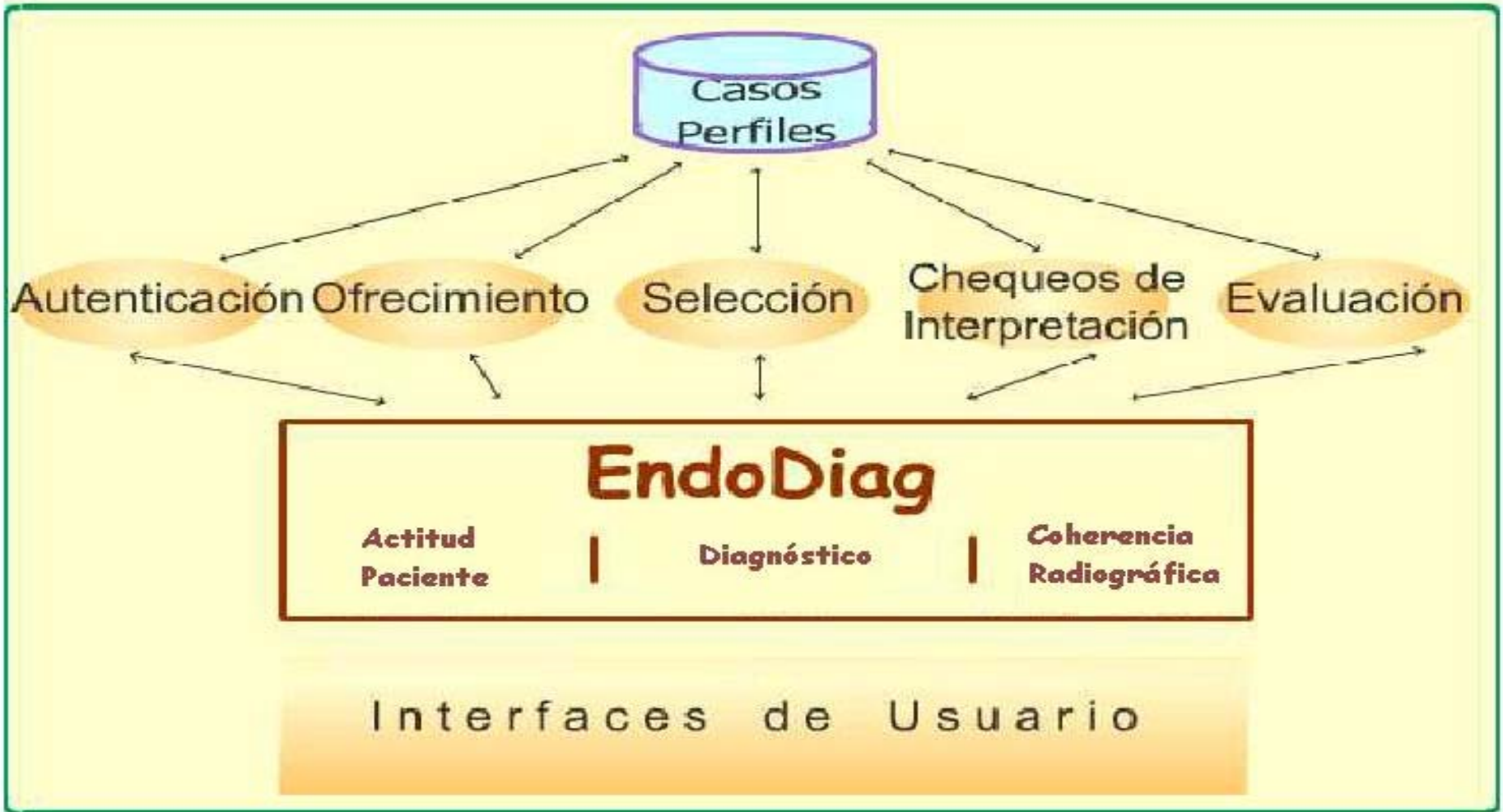


EndoDiagII como herramienta en el proceso de enseñanza-aprendizaje

Endodiag II (A. Torres)



Endodiag II

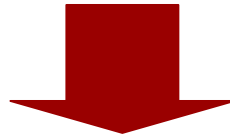


Conclusiones EndoDiag II

- La presencia de la base de casos facilita el uso del sistema.
- La inclusión de perfiles de usuarios favorece la enseñanza personalizada del proceso del diagnóstico.
- Necesidad de trasladarlo a una plataforma que sea accesible via Internet lo que permitiría una supervisión on-line del alumno.

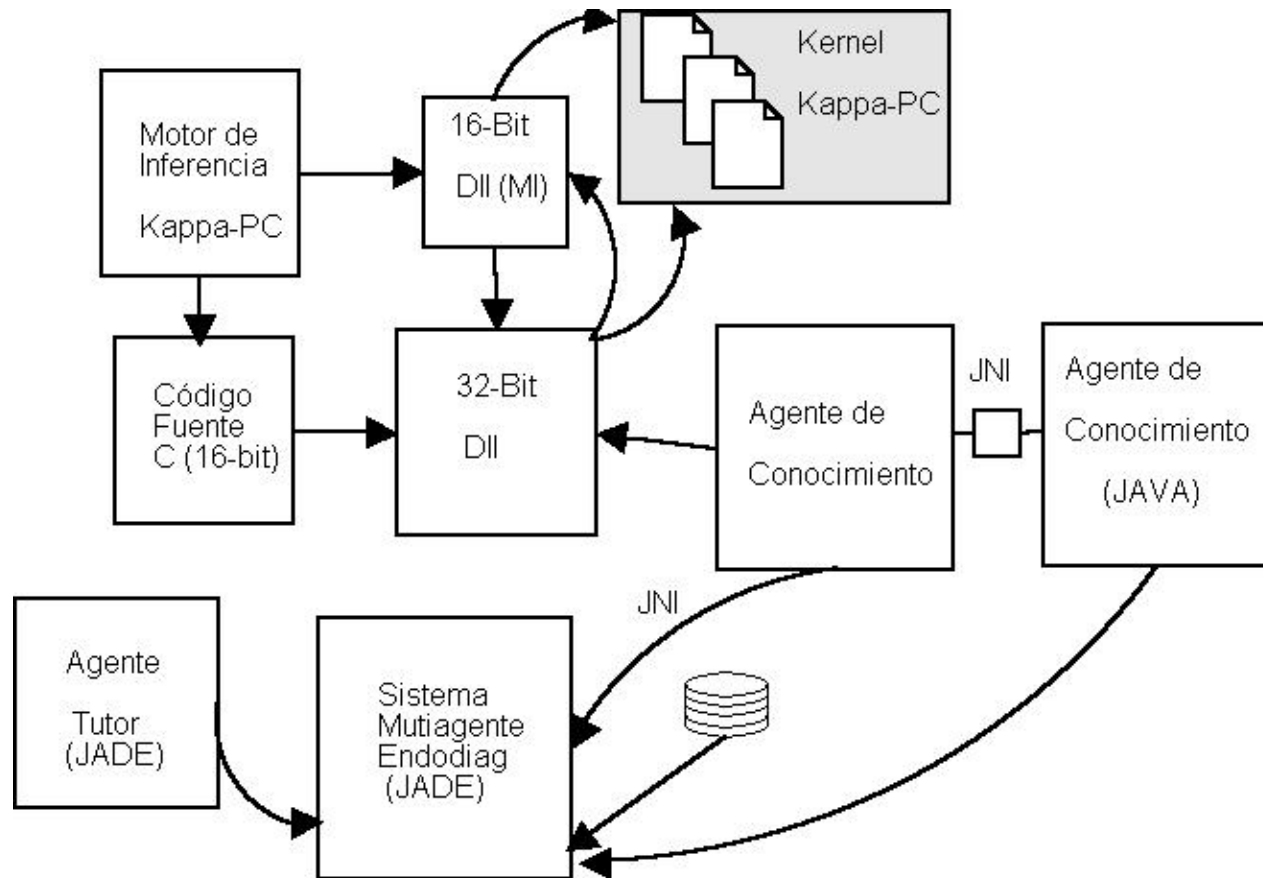
Endodiag+ :

- Sistema multiagente .
- Sistema orientado a la web.
- Sistema que mantenga la funcionalidad requerida en EndoDiag y Endodiag II.



Se necesita un cambio de arquitectura

Endodiag+ : Primera aproximación



Trabajos futuros:

- Desarrollar la herramienta Gkappa, para superar las limitaciones de Kappa-PC.
- En primera instancia, sólo se incorporaría la funcionalidad que poseen sus versiones anteriores

Un poco de motivación técnica:THUNK

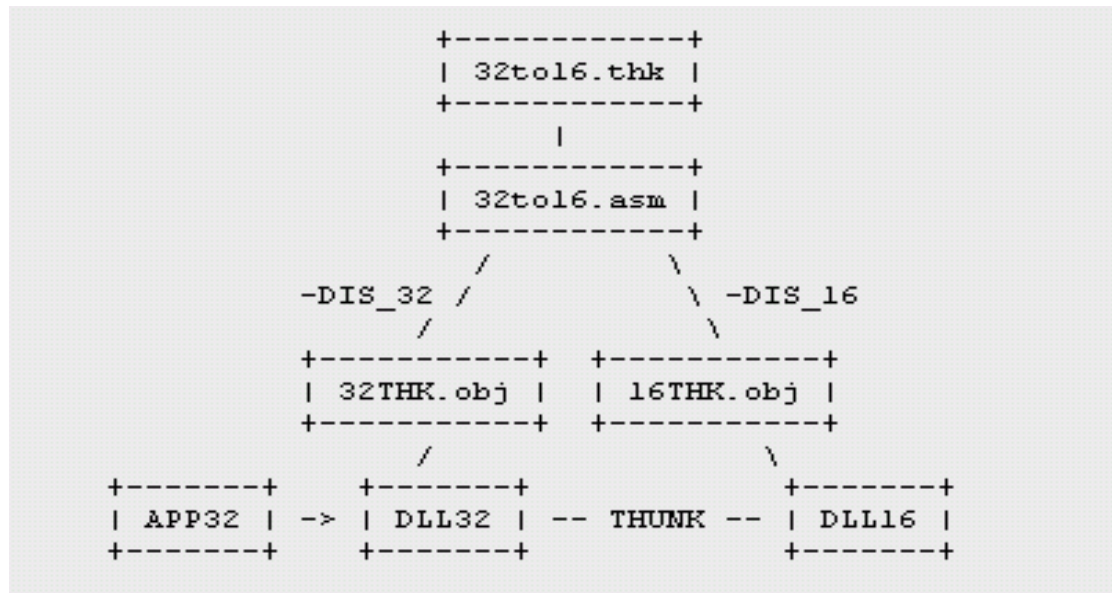
Existen tres técnicas distintas :

- FLAT :Win95-Win98-WinMe
- GENERIC:WIN2000-NT,XP,
- DIRECT:ídem flat-.Por lo tanto no son compatibles!!!

Un poco de motivación técnica:THUNK

- Un thunk consiste en 2 dll una de 32 y otra de 16 , en las llamadas de 32-16 la dll de 32 llama a una función exportada en la dll de 16 bits.Por lo tanto necesitamos un compilador de 16 bits , uno de 32 bits , el compilador thunk y el MASM por ej.. , este ultimo para ensamblar los .asm generados por el compilador thunk.

Un poco de motivación técnica: THUNK



- 1-Crear un thunk script .thk
- 2-Compilarlo con el “THUNK” ,generando .asm
- 3-Ensamblar para generar los .OBJ
- 4-Los módulos obj se deben linkear a las respectivas dlls.

Proyecto GKappa

Objetivos del proyecto:

- Utilización del código abierto obtenido y lograr una migración del producto a una tecnología de 32 bits.
- Generación de herramientas que asistan la migración.
- Portabilidad a sistemas Linux.
- Documentar y publicar el proyecto a la comunidad.

Proyecto GKappa

Beneficios del desarrollo de este proyecto

- Se obtendría en una primera fase el motor de inferencia totalmente en tecnología de 32 bits y con licencia de código abierto (gpl). Esto facilitaría el desarrollo casi-automático de sistemas de alto nivel que usen lo utilicen (ahora es muy artesanal)
- Se brindaría este código a toda la comunidad GNU vía la utilización de la web y sería fomentada por la cátedra, lo que proporcionaría el acercamiento de esta "comunidad".
- Se brindaría un software totalmente desarrollado en la facultad y mantenido vía web por la misma (único en nuestra carrera)
- Se estaría cumpliendo con la migración de un software que ha sido discontinuado de producción comercial.
- Se obtendría un software modular de código abierto ,para que cualquier alumno lo adapte a sus necesidades.

Proyecto GKappa

Diferencias entre las arquitecturas

- declaraciones de procedimientos de ventana
- declaraciones de punteros (near y far)
- tipos de datos
- mensajes
- llamadas a las api del sistema (las api estan desactualizadas ,no son las mismas en los 2 sistemas sol:thunking ¿? no obviamente)
- -funcion winmain (cambia la declaración y/o definición ¿?)

Proyecto GKappa

Metodología de trabajo

- La migración de tecnologías requiere de un análisis completo del código fuente del software (sol: herramientas que automaticen la búsqueda de problemas), por lo tanto es necesario definir las técnicas a seguir en este camino, por ejemplo podríamos adoptar un camino de tipo top-down (porque ?)(seria obtener los problemas utilizando un compilador de 32-bits y comenzar a tratarlos según ciertas reglas.

Proyecto GKappa

Ejemplo código Kappa :

(Presente en el código de generación de la dll ->kaprule.dll)

- **short NEAR** RulePriority (ITEMID idRule)
- {
- LPRULE lpRule;
- **short** sRet;
- if ((lpRule = (LPRULE) KppGetItem (RULE, idRule)) == NULL)
- return 0;
- sRet = PRIORITY (lpRule);
- KppReleaseItem (RULE, idRule);
- return sRet;
- }

Proyecto GKappa

Posible código GKappa :

(Presente en el código de generación de la dll ->kaprule.dll)

- **short** RulePriority (ITEMID idRule)
- {
- LPRULE lpRule;
- **short** sRet;
- if ((lpRule = (LPRULE) KppGetItem (RULE, idRule)) == NULL)
- return 0;
- sRet = PRIORITY (lpRule);
- KppReleaseItem (RULE, idRule);
- return sRet;
- }

Proyecto GKappa

Posible código GKappa

Near se remueve debido a que en en tecnología de 32 bits ,far y near son removidos debido a la arquitectura.

Short es de 16 bits tanto en la tecnología de 16 bits como en la de 32

Si en vez de short tendríamos int si ; tendríamos una complicación debido a que en 32 es de 32 bits , no así en la de 16 que es de 16 bits. Y NULL?

Proyecto GKappa

Estado del arte

- Finalización de la metodología a adoptar para realizar la migración, punto visto anteriormente. (investigación)
- Construcción de herramientas para la detección de problemas.
- Aislar el motor de inferencia, para implementar el primer paso de la migración.
- Generación del sitio y la documentación.

Proyecto GKappa

Trabajo a futuro

- Completar la migración incluyendo la del compilador de KAPPA.
- Portabilidad de Kappa a sistemas Linux.

Proyecto GKappa

Bibliografía

- 1. Departamento de Sistemas e Informática Facultad de Ciencias Exactas, Ingeniería y Agrimensura – UNR. Proyecto GKAPPA.
- 2. Casali A., Corti R., D'Agostino E., Biga A., Siragusa M., Aciar S. Sistema de apoyo al aprendizaje diagnóstico: de Endodiag a un sistema multiagente.
- 3. Casali A., Corti R., D'Agostino E., Siragusa M. Sistema Basado en Conocimiento de Apoyo al Diagnóstico de la Patología Pulpar y Periapical. Anales 31 JAIIO, Actas de SIS – ISSN 1666-1125, vol 5, pp 192-196. Santa Fe, 2002.
- 4. Casali A., Corti R., D'Agostino E., and Siragusa M.. Herramienta tecnológica como apoyo al diagnóstico endodóntico. Electronic Journal of Endodontics Rosario Ejer. ISSN 1666-6143 Año 3 - Volumen 1, 2004.
- 5. MSDN. www.msdn.com
- 6. “Undocumented Windows 2000 Secrets” - Sven V. Schreiber
- 7. “Windows 95 System Programming Secrets” - Matt Pietrek