

Una respuesta en un post...

Todo lo que voy a decir les sonará muy raro pero para mí son cuestiones fundamentales. Cuando en el 1999-2001 emprendimos con .NET, Windows Servers, SQL, SharePoint, CRM, Biztalk y Commerce todos creían que estábamos locos (sobre todo mis muchos amigos javeros y velneadores). Que todos ellos me digan ahora si lo estaba, seguro que nos echamos unas risas. Estas tecnologías me han permitido vivir dignamente los últimos diez años y nos permitirán crecer en los próximos cinco sin problemas. No era sencillo entonces tenerlo claro.

Piensen lo que quieran y exprésenlo aquí y donde deseen. A mí me encanta escuchar vuestras opiniones, pero no esperen que cambie de parecer simplemente por su visión del mercado y de sus específicos trabajos. Mi experiencia profesional es suficiente para tener claras las tendencias del mercado en lo técnico (sobre todo a alto nivel), bueno lo dicho “viva la democracia”.

Antes de empezar quiero dejar claro que el término “antiguo” lo aplico al punto de vista tecnológico. Que una plataforma sea “antigua” (tecnológicamente hablando) no quiere decir que no se use (véase AS/400, Cobol o Clipper). Y ahora al grano...

Hay que ser realista, la nueva era será inicialmente un diseminado de plataformas (no habrá un solo ganador);

- Microsoft con .NET y Azure,
- Google con Google Apps, Android y ChromeOS,
- Apple con iOS y sus dispositivos,
- Nokia con sus amigos y Meego...

Todos ganarán... Por lo que hay que tener en cuenta varias cuestiones fundamentales:

No pueden abordar de golpe todo el mercado.

La tarta se repartirá inicialmente entre todos.

La cuestión fundamental vendrá cuando la tarta ya esté repartida y ahí se librarán las batallas más importantes (lo de ahora es el aperitivo).

Desde nuestra visión, **los mejores lenguajes para los nuevos paradigmas tecnológicos son C# y Java, el padre de todos ellos es C++**. A tener en cuenta Ruby y F#.

Las mejores plataformas de desarrollo GTK y Qt. El motivo de esta afirmación es

básicamente que las aplicaciones necesitarán correr en dispositivos distintos y de prestaciones limitadas.

Las **plataformas** actuales; Unix, Windows, Linux, Embedded, MacOSX, RIM, Symbian, Windows Mobile y Windows CE están posicionadas en grandes mercados. Ahora tenemos nuevos participantes de peso Android, Windows Phone 7, iOS... **¿Qué tienen todos en común?, C++** (ni si quiera html5) incluso todas las consolas...

En un futuro próximo la mayoría añadirá **html5**. **El futuro pasa por html5 y C++**, no tengo ninguna duda.

Respecto a las plataformas **GTK y Qt** me quedo especialmente con Qt por al apoyo empresarial de **Nokia+Intel+Amd+Auto+Novell+... y de la comunidad Linux**. También por QML y HTML5 es el camino directo al éxito de Qt (nada de rodeos). **Qt es una realidad en al menos Maemo, Meego, Unix, Windows, Linux, Linux Embedded, MacOSX, Symbian, Windows Mobile y Windows CE... tiene soporte completo en todos ellos ¿os parece poco?... .**

NET es Windows (vale también es Linux con Mono.net).

Java es una realidad en la mayoría de las plataformas enumeradas pero el soporte no es completo para todas.

Java y .NET son plataformas “antiguas” porque sus diseños se basan en lenguajes intermedios compilados. Este diseño las hace pesadas en ejecución e inadecuadas para esta nueva era tecnológica (véase [multitenancy](#)). A día de hoy son mayoritarias en aplicaciones cliente/servidor y lo seguirán siendo a largo plazo, de esto no tengo ninguna duda, pero no abordarán los nuevos paradigmas tecnológicos como lo hicieron hasta la fecha, **vean como .NET está evolucionando** de Windows forms->ASP.NET->Ajax->WPF->Silverlight->HTML5 (<http://es.engadget.com/2010/10/31/microsoft-apuesta-por-html5-y-retira-silverlight-de-su-estrategi/>). **Estudien en detalle Azure y entenderán esto que digo.**

De Java hablo menos, porque hace mucho tiempo que la deseché como plataforma de desarrollo para el futuro. Más de un javero seguirá pensando que estoy loco, bueno es un punto de vista.

Ahora el turno de los híbridos (generadores de código y plataformas intermedias). **WinDev/WebDev/Mobile** es una plataforma “antigua” ¿por qué versión va?, es pesada y tienen matices que hacen que la multiplataforma no sea completa. Para hacer una aplicación multiplataforma la tienes que hacer con múltiples tecnologías HTML+Ajax por un lado, por otro en cliente/servidor y por último en mobile (tres plataformas para un solo fin) y sinceramente de esta forma el resultado deja bastante que desear (en costes de factor humano). No han parado de dar tumbos (sigo a Windev desde su versión 8 y me cansé de escuchar cambios de rumbo). Lo bueno es que arrastra un conocimiento y comunidad muy importantes.

Servoy está mejor posicionada, pero no deja de ser Java y como apunté anteriormente

considero que es “antiguo”. Siento que esta afirmación ofenda a algunos, pero es mi parecer desde hace mucho tiempo atrás. Evalué en detalle estas plataformas y no son para nosotros, simple y llanamente. Si os gustan no lo discuto, cada uno decide profesional y empresarialmente su camino.

Ya sé que a día de hoy todas estas reflexiones suenan raro como en el pasado nuestra apuesta por .NET y sus productos empresariales... El tiempo nos dio razones de peso para usar .NET y sus productos empresariales. Ahora queremos incorporar Velneo a nuestra apuesta tecnológica.

Por cierto, si uno se cae que se levante. No me fio de las empresas que no se cayeron al menos tres veces en su vida. Si esta es una de ellas no hay problema, tenemos un buen colchón para ello. Pero mientras tanto nuestro aliento será para Velneo, Qt y Nokia.