

## RESUMEN

Ingeniero en Informática. Puesto actual: Jefe de Proyecto.  
9 años de experiencia en desarrollo de proyectos.  
Habilidades técnicas y organizativas.

## TRAYECTORIA PROFESIONAL

- **CENTRO INFORMÁTICO CIENTÍFICO DE ANDALUCÍA (CICA).**  
2 años. Becario.  
Funciones:
  - Instalación de software Digital VAX/VMS y configuración de servidores.
  - Soporte al personal docente e investigador de la Universidad de Sevilla en la utilización de software.
- **ASESORÍA Y CONSULTORÍA S.A.** Dpto. de Banca Electrónica y Medios de Pago.  
3 años. Posición alcanzada: Programador-Analista.  
Funciones:
  - Programación.
  - Apoyo a la fase de análisis de nuevas aplicaciones de banca.
- **PROMAINSUR S.A.**  
5 años. Posición alcanzada: Jefe de Proyecto.  
Funciones (actuales):
  - Dirección de proyectos.
  - Investigación del estado actual de la tecnología y sus aplicaciones en futuros proyectos.
  - Impartir formación en las tecnologías que domino especialmente bien.
  - Asesoramiento técnico a la dirección en la toma de algunas decisiones.

## FORMACIÓN ACADÉMICA

- Diplomatura en Informática: Escuela Universitaria Politécnica (Universidad de Sevilla).
- Ingeniero en Informática: Facultad de Informática (Universidad de Sevilla).

## HABILIDADES

UML, Rational Rose, UML Studio.  
CVS.  
Lenguajes: SQL, Perl, C, C++ y Java.  
Gestores de bases de datos: Teradata (NCR) y MySQL.  
Apache y tecnologías web relacionadas.  
Comunicaciones: certificación Cisco CCNA (en preparación).  
Tarjetas microchip. Especificaciones EURO6000. Software embebido en dispositivos.  
Inglés.

## DATOS PERSONALES

Nacido el 14 de Julio de 1.966 en Badajoz. Español. Casado.

## Mi historial de proyectos realizados

### Proyecto AIP ( Mar/1994 - Oct/1994 )

Cliente: *Caja de Ahorros El Monte (Dpto. Auditoría y Seguridad)*

Puesto: *Programador*

Descripción: Desarrollar una aplicación para auditar varios aspectos del sistema de información de El Monte.

Este fue un proyecto interno del Dpto. de Auditoría y Seguridad (que después fue dividido en Dpto. de Auditoría y Dpto. de Seguridad). El departamento realizó el análisis y yo la programación de la aplicación.

### Proyecto TEJARES ( Sep/1994 - Mayo/1996 )

Cliente: *CajaSur, IBM*

Puesto: *Programador C/C++, Instalador*

Descripción: Desarrollar la aplicación transaccional de un nuevo cajero (el IBM 4782) que en aquella época IBM incorporó a su gama de productos de medios de pago.

Instalar la aplicación en cajeros de CajaSur (en Sevilla, Morón de la Frontera, Isla Cristina, Baeza, Córdoba, Bélmez, Jaén y Almería).

Formé parte del equipo de desarrollo de la aplicación y después del de instalaciones.

### Proyecto ELFOS ( Mar/1996 - Jun/1996 )

Cliente: *Bull*

Puesto: *Tester*

Descripción: Comprobar el correcto funcionamiento del cajero Bull MiniCash con la aplicación de cajeros de El Monte:

Diseñé un conjunto de tests para el cajero Bull MiniCash que cubrieran todos los dispositivos del cajero y todas las transacciones de la aplicación de cajeros de El Monte.

Ejecuté cada test sobre un cajero NCR 5386 y también sobre el Bull MiniCash (ambos con la aplicación de El Monte instalada) para comparar ambos resultados.

Documenté el resultado de cada prueba en cada cajero y redacté un documento de conclusiones explicando el grado de compatibilidad del Bull MiniCash al ejecutar la aplicación de El Monte.

### Proyecto ONCE ( Ago/1996 - Sep/1996 )

Cliente: *Caja de Ahorros El Monte*

Puesto: *Programador*

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

Descripción: Modifiqué la aplicación de cajeros de El Monte para dotarla de los mecanismos necesarios que permitieran su uso por invidentes: mensajes hablados, tonos sonoros con significado, ...

## Proyecto SCAP ( Feb/1997 - Dic/1997 )

Cliente: *Diputación Provincial de Sevilla*

Puesto: *Programador-Analista*

Descripción: Implantar, en el edificio de la Diputación, un Sistema de Control de Accesos y Presencia basado en tarjetas microchip y terminales de lectura / grabación de tarjetas (22 terminales y aprox. 4000 tarjetas).

Mi contribución consistió en:

Diseñar la estructura de datos de la tarjeta (que se graba en el chip al personalizar la tarjeta).

Programar los terminales de control de acceso y diseñar y programar la aplicación de control de los mismos.

Realizar el análisis del resto de las aplicaciones que se desarrollaron en el proyecto.

Dar soporte a los desarrolladores de las restantes aplicaciones, en cuanto a software base y tecnología de tarjetas.

## Integrar la tarjeta EURO6000 en la aplicación de terminales de El Monte ( Sep/1997 - Feb/1998 )

Cliente: *Caja de Ahorros El Monte*

Puesto: *Programador-Analista*

Descripción: El objetivo del proyecto era que los clientes de El Monte pudiesen utilizar su tarjeta EURO6000 para realizar transacciones (consulta de saldo, carga de efectivo, ...) en las oficinas de la Entidad. Para ello, El Monte compró una partida de lectores / grabadores de tarjeta tipo sobremesa, para instalar uno o dos de ellos en cada una sus oficinas.

Se necesitaba dotar a la aplicación terminales (la que los empleados de oficinas utilizan para realizar sus tareas diarias mas comunes), de la capacidad de realizar transacciones con una tarjeta EURO6000. En definitiva, integrar la tarjeta EURO6000 en la aplicación de terminales.

El método de integración consistió en desarrollar varias librerías de software que permitieran a la aplicación realizar transacciones de tarjeta EURO6000 (consulta de saldo, carga de parámetros, ...). Las librerías se encargan de gestionar el dispositivo lector / grabador, de interaccionar con el sistema operativo de la tarjeta y de generar / analizar los mensajes de transacción que se intercambian con la aplicación de CECA.

Mi misión consistió en desarrollar las librerías para sistema operativo OS/2 y Windows 95.

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

## Bono turístico de la ciudad de Granada ( Jun/1998 - Ene/1999 )

Cliente: *Caja de Ahorros General de Granada*  
Puesto: *Programador-Analista*  
Descripción: El objetivo del proyecto fue emitir una tarjeta microchip de bajo coste y de un solo uso, que se pudiese adquirir en agencias de viajes, hoteles, ... , de forma que la compra de la tarjeta llevase aparejado el prepago de un conjunto de ofertas turísticas asociadas a la ciudad de Granada.

## Proyecto VILLASIS ( Nov/1998 - Sep/1999 )

Cliente: *Caja de Ahorros El Monte*  
Puesto: *Jefe de Proyecto*  
Descripción: Desarrollar un control de accesos y presencia al edificio Villasis de El Monte, basado en tarjetas microchip y terminales de lectura / grabación de tarjetas.

Lo novedoso de este proyecto respecto al de la Diputación fue que en este desarrollamos una aplicación de terminales completamente nueva, fácilmente migrable a cualquier tipo de terminal e independiente del tipo de tarjeta utilizada. También reescribimos completamente el software de control de terminales para que el sistema pudiese configurarse tanto para funcionamiento *online* como *offline* (en *offline* cada terminal guarda los accesos producidos en él, y un proceso posterior de descarga los recupera de cada terminal, mientras que en *online* el acceso se guarda en tiempo real en la base de datos del sistema).

## Proyecto ABC ( Jun/1999 - Ago/1999 )

Cliente: *ABC Prensa Española*  
Puesto: *Jefe de Proyecto*  
Descripción: Implantar un sistema de control de accesos y presencia de similares características al instalado en el edificio Villasis.

## Proyecto UNIVERSIDAD de CORDOBA ( Sep/1999 - Sep/2000 )

Cliente: *Universidad de Córdoba*  
Puesto: *Jefe de Proyecto*  
Descripción: Realmente, estos fueron dos proyectos:

Basándose en el hecho de que el carnet universitario es una tarjeta EURO6000 (emitida por CajaSur), implantar un sistema de control de accesos a parkings, en el campus de la Universidad.  
Los estudiantes y profesores utilizan la tarjeta para que el sistema de control de accesos les autorice la entrada al parking.

También basado en la tarjeta EURO6000 de CajaSur, implantar un sistema de control de presencia para el personal de la Universidad.  
En cada facultad hay uno o varios lectores / grabadores de tarjeta con módem, conectados a la red telefónica interna de la Universidad. Una aplicación centralizada controla los dispositivos via acceso telefónico.

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

## Proyecto Cybercafes ( Dic/1999 - Feb/2000 )

Cliente:

Puesto: *Jefe de Proyecto*

Descripción: Desarrollar un sistema de securización de acceso a ordenadores personales y pago automático en función del tiempo de uso, ambos basados en tarjetas microchip y el sistema de securización de Windows NT (MSGINA.DLL). Se pretendía que el sistema de securización sirviese también para entidades financieras, organismos públicos, ...

Este proyecto no llegó a pasar de la fase de investigación por problemas de comercialización del producto que se pretendía desarrollar.

## Proyecto PINEDA ( Mar/2000 - Oct/2000 )

Cliente: *Club Pineda (Sevilla), Caja de Ahorros El Monte*

Puesto: *Jefe de Proyecto*

Descripción: El proyecto consistió en cambiar la tarjeta identificativa de los socios del club por una tarjeta EURO6000 emitida por El Monte, y basándose en el hecho de que cada socio tiene una EURO6000, desarrollar un conjunto de aplicaciones basadas en la tarjeta.

Las aplicaciones desarrolladas permiten:

- Control de presencia y accesos al recinto del club.
- Pago de servicios.

Este proyecto fue pionero en utilizar la tarjeta financiera EURO6000 para otros usos como control de accesos y presencia además de como medio de pago.

## Bono Turístico de Mallorca ( Abr/2000 - Jun/2000 )

Cliente: *Iniciativas Turísticas S.A. (INITUR)*

Puesto: *Jefe de Proyecto*

Descripción: Desarrollar un sistema de prepago de espectáculos y atracciones turísticas basado en tarjetas microchip y dirigido al sector turístico de Mallorca.

## Proyecto SERVICIO CANARIO DE SALUD ( Ago/2000 - Sep/2000 )

Cliente: *Servicio Canario de Salud*

Puesto: *Jefe de Proyecto*

Descripción: Implantar un sistema de control de accesos basado en tarjetas microchip y terminales lectores / grabadores, en el Servicio Canario de Salud.

Lo novedoso de este proyecto fue el que los terminales de control de accesos están instalados en edificios situados en dos islas distintas, e incorporan un módulo que les permite funcionar como terminales TCP/IP.

## Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

### Proyecto Universidad de Alcalá de Henares ( Sep/2000 - Nov/2000 )

Cliente: *Intelligent Data S.A., Univ. de Alcalá de Henares*  
Puesto: *Jefe de Proyecto*  
Descripción: Implantar un sistema de Control de Accesos en la Universidad de Alcalá de Henares, basado en la tarjeta universitaria.

### Proyecto e-DOSSIER ( Sep/2001 - Feb/2002 )

Cliente: *Caja de Ahorros El Monte*  
Puesto: *Consultor / Jefe de Proyecto*  
Descripción: Diseñar una plataforma tecnológica que:

- Implemente el concepto de **multicanal** en la Entidad.
- Proporcione los API,s y mecanismos necesarios para la gestión de **expedientes electrónicos** de cualquier tipo.
- Facilite el desarrollo de futuras aplicaciones de expediente electrónico permitiendo a éstas utilizar los mecanismos y servicios ya implementados en la plataforma.

El resultado del proyecto fue un documento que especifica, en UML, el diseño de la plataforma, (UML o *Unified Modeling Language*, es un lenguaje para modelizar, especificar y documentar sistemas de información).

### Encaje de Efectivo en Oficinas de Entidades Bancarias ( Mar/2002 - Feb/2003 )

Cliente: *Caja de Ahorros El Monte*  
Puesto: *Jefe de Proyecto*  
Descripción: Desarrollar un sistema que gestione el encaje de efectivo en oficinas de entidades bancarias, y minimize el coste total del encaje para la Entidad.

El encaje de efectivo es la cantidad de efectivo que una oficina necesita para su funcionamiento normal. Esta cantidad de efectivo no es la misma para todas las oficinas, ni en una misma oficina, en todas las épocas del año.

Por otra parte, los encajes de las oficinas tienen asociados unos costes (interés del dinero, trasado de fondos, seguros, ...) que hay que minimizar.

El sistema desarrollado como resultado del proyecto, minimiza los costes asociados a la gestión del efectivo. Se basa en modelos matemático-estadísticos que realizan predicciones de la cantidad de efectivo que va a necesitar cada oficina en los próximos días, a la vez que permite a las oficinas y a las cajas centrales realizar simulaciones del proceso de gestión del encaje, para ser aconsejadas sobre las necesidades de efectivo futuras.

El sistema se compone de un conjunto de aplicaciones web (Java, WebSphere) que interaccionan con una base de datos Teradata.

## Las herramientas que utilizo

Es evidente que en un proyecto software es necesario el uso de multitud de herramientas, cada una de ellas de un tipo diferente, es decir, cada una de ellas especializada en realizar una tarea específica en una o varias de las fases del proyecto.

**Mi objetivo ha sido siempre disponer de al menos, una herramienta de cada tipo, de forma que entre todas ellas cubran todas las fases y tareas más comunes de un proyecto software.**

A continuación explico los tipos de herramientas que, en mi opinión, son necesarios en la mayoría de proyectos software, y que herramientas de cada uno de estos tipos utilizo yo:

- Una herramienta, basada en tecnología UML, para **especificar y modelizar software**.

### Rational Rose

La herramienta de modelado UML por excelencia, como no podía ser de otra forma: los *inventores* del UML (Rumbaugh, Jacobson y Booch) colaboraron estrechamente con Rational (antes de ser adquirida por IBM), en el desarrollo de toda su familia de herramientas UML.

### UML Studio

Admite todos los tipos de diagramas de UML, se pueden crear nuevos símbolos y genera código para Java y C++, y su relación precio/capacidades es inmejorable.

- Una herramienta que permita el **trabajo concurrente** de todas las personas que forman parte del equipo de desarrollo, sin importar el número de personas que lo formen ni su distribución geográfica, y que además, permita **manejar repositorios de códigos fuente**.

### CVS

CVS (Concurrent Versions System) es sin ninguna duda el mejor software de este tipo.

Además de realizar a la perfección las tareas mencionadas (mas otras muchas que por espacio no puedo mencionar aquí), tiene tres importantes ventajas sobre otros productos similares: funciona en modo mezcla (otros funcionan en modo bloqueo), es multiplataforma (disponible en UNIX/Linux y Windows) y se distribuye bajo licencia GNU-GPL.

Ideal para subcontratar partes de un proyecto.

- Una herramienta que permita a los programadores **interaccionar con el repositorio**, independientemente del tipo de plataforma en la que éste resida.

### Cliente por defecto de CVS

Cliente de línea de comando que se suministra en la distribución de CVS

### WinCVS

Un cliente de CVS con una interfaz gráfica a base de ventanas. He traducido el manual de este producto al castellano.

Para mostrar las diferencias entre dos releases distintas del mismo fichero requiere una herramienta externa que realice esta tarea. Yo utilizo [WinDiff](#).

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

- Una herramienta para **generar la documentación** del software a partir de los códigos fuentes, a ser posible en varios formatos (RTF, PDF, HTML, ...).

[pod2html](#), [pod2pdf](#), [pod2latex](#), [pod2rtf](#), [pod2man](#), [pod2text](#), [pod2chm](#)

pod2html es una herramienta que se suministra en las distribuciones de Perl. Extrae de un archivo fragmentos de documentación escritos en lenguaje POD, y genera otro fichero con el resultado de traducir los fragmentos a HTML. Existen versiones para convertir de POD a cada uno de los formatos de archivo más comunes: pdf, LaTeX, rtf, man, texto, ...

[POD](#) (Plain Old Documentation) es un formato de documentación bastante *neutro* (los documentos escritos con él se pueden generar en cualquier otro formato) y *sencillo* (POD es texto con algunas etiquetas). Cada una de estas utilidades está también disponible como un API de programación que permite generar la documentación programáticamente en el formato correspondiente, así es posible automatizar la generación de documentación en múltiples formatos.

[JavaDoc](#)

Extrae documentación de códigos fuente en Java.

Mi propia herramienta

Todavía en desarrollo. Integrará en una sola herramienta JavaDoc y las otras que se distribuyen con Perl (pod2html, pod2pdf, ...). Se distribuirá bajo licencia GNU-GPL.

- Un **servidor de bases de datos**.

[Teradata](#)

Sin duda el mejor RDBMS para la implantación de un Data Warehouse.

[Oracle](#)

Muy extendido, con muchas e interesantes extensiones al estandar SQL, y dispone de una familia de excelentes productos de desarrollo, basados en el servidor de bases de datos.

[MySQL](#)

Para aplicaciones de bajo coste.

- Una herramienta cliente que permita **ejecutar SQL**, a ser posible independientemente de cual sea el servidor de bases de datos.

[WinDDI](#)

Cliente de Teradata RDBMS. Para máquinas Windows.

[MySQL Control Center](#)

Cliente de MySQL. Es una herramienta que pueden usar tanto programadores como administradores: proporciona una interfaz cómoda para escribir y ejecutar código SQL, a la vez que permite administrar cualquier número de servidores de bases de datos MySQL. También para máquinas Windows.

[Emacs con módulo SQL](#)

El módulo SQL de Emacs colorea la sintaxis a la vez que uno escribe. Además, se puede ejecutar el código SQL a la vez que se escribe, y ver los resultados. Todo ello dentro del mismo entorno.

En la configuración del módulo SQL se especifican los parámetros necesarios para establecer la conexión con el servidor de bases de datos.

Este cliente se puede usar en cualquier plataforma donde funcione Emacs, que son prácticamente todas (todos los UNIXes, Windows, VMS, ...). Para mí esto es una característica muy importante porque me permite trabajar en varias plataformas sin necesidad de utilizar herramientas específicas en cada una de ellas.

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

## DBI Shell

Es un cliente de línea de comando que proporciona conectividad con cualquier base de datos vía DBI (interfaz de conexión con bases de datos que se usa en el mundo Perl).  
Por estar basado en Perl, este cliente es también multiplataforma. Se puede usar con una interfaz más cómoda utilizándolo como cliente del módulo SQL de Emacs.

- **Lenguajes de programación.**

## Perl

Lo utilizo para todo, y en cualquier plataforma. No he encontrado nada que no se pueda programar con Perl. Desde administración de servidores hasta interfaces gráficas de usuario.  
La [CPAN](#) es el mayor repositorio de librerías (*modules* en terminología Perl), de uso público.

## Java

Lo utilizo para aplicaciones en las que el cliente pide explícitamente el uso de este lenguaje y otras tecnologías relacionadas con él.

## C/C++

Lo utilizo para desarrollar software embebido en dispositivos y aplicaciones o partes de aplicaciones donde la velocidad es un factor esencial.

- **Un entorno de desarrollo** que cumpla, al menos, con las funciones básicas: escribir código fuente y documentación, buscar cadenas de caracteres entre todos los ficheros de un proyecto, mostrar las diferencias entre dos ficheros, depurar el código, ejecutar makefiles, y si el lenguaje lo requiere, compilar y enlazar.  
Algunos entornos de desarrollo, además de realizar estas funciones básicas tienen integradas algunas herramientas mencionadas en puntos anteriores, por lo que en caso de utilizarlos, se reduce en número de herramientas necesarias.

## Emacs

Emacs es, además de un potente editor de códigos fuente para desarrolladores, un completo entorno de desarrollo modular. Colorea la sintaxis de cualquier lenguaje (Java, Perl, C/C++, Ada, Python, SQL, HTML, SGML, XML, ...), existe un módulo que proporciona una interfaz de alto nivel con cualquier sistema de control de versiones (en particular CVS), y se pueden realizar multitud de funciones sin salir del entorno: compilar y enlazar, comparar ficheros, buscar cadenas de caracteres en el árbol de directorios de un proyecto, ...

Uno instala los módulos que necesita, y cuando necesita nuevas funcionalidades puede instalar nuevos módulos que las realicen. De esta forma, uno se puede configurar su propio entorno de desarrollo. Es multiplataforma, lo que me permite cambiar de una plataforma a otra y seguir utilizando el mismo entorno de desarrollo.

## WebSphere Studio, Studio Site Developer

Completo entorno de desarrollo para Java. Entre otros muchos componentes, tiene integrado un cliente de CVS, lo que permite trabajar con un repositorio de fuentes CVS sin abandonar el propio entorno de desarrollo.

Puede trabajar con servidores de aplicaciones WebSphere (IBM) y Tomcat (este último se distribuye bajo licencia GNU).

- **Una herramienta para generar analizadores léxicos.**

## Perl y expresiones regulares

Las expresiones regulares de Perl permiten codificar cualquier analizador léxico solo con este lenguaje, sin tener que recurrir a herramientas externas.

## Flex

Flex es Lex (el generador de analizadores léxicos típico del mundo UNIX), pero con licencia GNU y multiplataforma. Genera código C o C++

- Una herramienta para **generar analizadores sintácticos**.

## Bison

Bison es Yacc (el generador de analizadores sintácticos típico del mundo UNIX), pero con licencia GNU y multiplataforma. Genera código C o C++.

## Yapp

Yapp (Yet Another Perl Parser compiler) es también un Yacc pero desarrollado como una librería de Perl (Parse::Yapp) por Francois Desarmenien. Por supuesto, genera código Perl, se distribuye bajo licencia GNU y funciona en cualquier plataforma donde funcione Perl.

- Una **librería de componentes (*widgets*)** para desarrollar Interfaces Gráficas de Usuario (GUI).

## Perl/Tk

Tk es una excelente librería de componentes (controles, widgets) para desarrollar interfaces gráficas de usuario, creada por el MIT. Tradicionalmente, la única forma de usar Tk era programando en lenguaje Tcl (Tcl/Tk).

Hasta que Nick Ing-Simmons creó Perl/Tk (thanks Nick). El trabajo de Nick consistió en integrar Tk en Perl, es decir, desarrollar una interfaz de Perl hacia Tk.

Con Perl/Tk se pueden desarrollar aplicaciones a base de interfaces gráficas de usuario y orientación a objetos, así como desarrollar nuevos widgets a partir de otros.

Perl/Tk se distribuye como un conjunto de librerías de Perl bajo licencia GNU-GPL, y esta disponible para casi cualquier plataforma, incluido Windows.

- Una herramienta que permita **testear el software** desarrollado.

No hay una solución universal al problema de testear aplicaciones, entre otras razones porque el software de test depende del tipo de la aplicación y de la plataforma. Por ejemplo, los mecanismos de test para una aplicación web no sirven para una aplicación de escritorio, y la misma aplicación de escritorio en plataformas distintas se testea con software diferente.

Lo que si hay son excelentes soluciones parciales. Por ejemplo, existen excelentes productos para testear aplicaciones .NET, o aplicaciones web basadas en Java y WebSphere.

## IBM Rational XDE Tester

Automatiza la ejecución de tests de regresión y funcionales para aplicaciones web desarrolladas en Java. Se integra con IBM WebSphere Studio.

## Test::Builder, Test::Harness, Test::More, Test::Simple

Son 4 librerías (módulos) de Perl para escribir tests de aplicaciones.

Perl es un lenguaje especialmente adecuado para escribir tests de aplicaciones servidoras (aplicaciones web, aplicaciones de bases de datos, ...), y por supuesto aplicaciones Perl.

- Una herramienta para **generar distribuciones** del software y los manuales.

## Wise Installer

Genera distribuciones de software para plataformas Windows.

## ExtUtils::MakeMaker

Una librería Perl que genera distribuciones de software. Funciona en cualquier plataforma donde funcione Perl.

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

- Un sistema de **distribución de software** de forma remota.

## [CPAN \(Comprehensive Perl Archive Network\)](#), [Cliente de la CPAN](#)

La CPAN es el sistema de distribución remota de software mas sencillo y eficiente que conozco. Un sistema de distribución remota de software necesita: un repositorio de paquetes software, un servidor que publique el material del repositorio y un cliente capaz de buscar, descargar e instalar paquetes.

La CPAN es esto mismo: el repositorio de paquetes es una estructura de carpetas en una máquina o en una base de datos, el servidor es el de FTP y HTTP, y el cliente se llama CPAN::Shell.

Instalando otros repositorios con la misma estructura de la CPAN, y permitiendo el acceso con los mismos protocolos, se puede usar el cliente de la CPAN aunque el repositorio no sea la propia CPAN sino otro similar. CPAN::Shell es un cliente de la CPAN basado en Perl. Entre otras funcionalidades permite buscar, examinar, instalar y generar distribuciones de software.

Como muchas librerías Perl, además de un API de programación, dispone de una interfaz de usuario a base de línea de comando. El API permite desarrollar procesos de mirroring y de actualización automática de versiones.

- Una herramienta que permita hacer un **seguimiento de bugs** detectados en el software.
- Una herramienta que permita **publicar y asignar tareas pendientes**.
- Una herramienta que **extraiga información estadística acerca de un proyecto** (tiempo empleado en el desarrollo, número de personas o perfiles de cada tipo que han participado, número y tamaño de los ficheros desarrollados, ...).
- Una herramienta que permita **medir el esfuerzo necesario** en el desarrollo de un proyecto.
- Una herramienta para **planificar proyectos**, asignando personas y recursos.
- Una herramienta para **controlar proyectos**, que permita hacerse una idea del grado de cumplimiento respecto a la planificación.
- Herramientas adicionales: **ofimática, compresores, navegador web, cliente de correo, cliente FTP, cliente Telnet y visualizadores** de los principales tipos de archivos (PDF, Postscript, ...).

Además, para determinados tipos de proyectos puede requerirse el uso de herramientas adicionales. Por ejemplo, en el desarrollo de aplicaciones tipo web:

- Un **servidor de aplicaciones web** o al menos, un **servidor HTTP**.

## [IBM WebSphere](#)

Un servidor de aplicaciones web basado en Java. El producto que hace las veces de servidor HTTP es Apache.

Se integra perfectamente con el entorno de desarrollo IBM WebSphere Studio

## [Tomcat](#)

Un servidor de aplicaciones web basado en Java. Tomcat es una ampliación de Apache. Licencia GNU.

## [mod\\_perl](#)

mod\_perl es un módulo (una expansión que añade nuevas funcionalidades) del servidor HTTP Apache.

Con mod\_perl instalado, un servidor Apache es un servidor de aplicaciones web basadas en Perl.

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

## [Apache](#)

El mejor y más extendido servidor HTTP.

- Una herramienta para **diseñar y ensamblar componentes HTML**.

## [Template Toolkit](#)

Es una herramienta de templates, válida para usar templates en cualquier tipo de aplicación, no solo HTML.  
Licencia GNU

- Una herramienta para **diseñar formularios**.

## [Adobe GoLive](#)

Es una excelente herramienta de diseño web, con funciones mucho más amplias que el diseño de formularios HTML, aunque yo la utilizo principalmente para esto.

- Una herramienta para **diseñar y aplicar hojas de estilo**.

## [TopStyle](#)

Aunque otras herramientas como GoLive también sirven para diseñar hojas de estilo, sin embargo, la herramienta más especializada en esta tarea es Top Style.

- Un **lenguaje de scripting** en el lado del cliente web.

## [JavaScript](#)

- Una **herramienta XML que automatice la confección de listados y gráficos**.

Uno de mis proyectos (a proyectar), es el desarrollo de una herramienta de este tipo.

- Una **herramienta XML para implementar interfaces de usuario**.

## [Struts](#)

- Una herramienta para **desarrollar portales sin tener que partir de cero**.

## [Slash](#)

Un producto para implementar un tipo especializado de portal denominado

- Una herramienta para **tratamiento de imágenes**.

Por ahora, el tratamiento de imágenes no es un campo que llame mi atención.

Por supuesto, para manejar la mayoría de estas herramientas es necesario conocer la tecnología que aplican. Por ejemplo, de nada sirve disponer de la mejor herramienta para modelar sistemas con UML si uno no conoce la tecnología UML.

En la [página de las tecnologías que domino](#) encontrará la lista de tecnologías en las que se basan estas herramientas.

Finalmente, he aquí los criterios que siempre he seguido, para elegir que herramientas incorporar a mi repertorio:

- Potencia.
- Portabilidad.
- Tipo de licencia.
- Amplia aceptación.

## Las tecnologías que domino

Por lo general, en una aplicación software conviven en armonía varias tecnologías (orientación a objetos, bases de datos, servidores web, ...). Cada una de estas tecnologías juega un papel específico, y casi todas están disponibles para los desarrolladores en forma de paquetes software, como las piezas de un puzzle.

El trabajo del desarrollador consiste en escribir un poco de software, a modo de pegamento entre las piezas. También hay huecos libres a los que ninguna pieza se adapta, por lo que es necesario fabricar una a medida, preferiblemente a partir de otra pieza ya existente.

Desde el punto de vista técnico, las claves del éxito de un proyecto software son: saber que conjunto de piezas es el mas apropiado para componer el puzzle, cuanta cantidad de pegamento será necesaria, que modificaciones necesitarán algunas de las piezas y como llevarlas a cabo de la forma más óptima. En definitiva, **conocer la tecnología**.

Mi reto permanente es:

**Dominar en profundidad un conjunto de tecnologías de uso frecuente en la mayoría de los proyectos software:**

- Tecnologías web.
- Tecnologías de comunicaciones.
- Bases de Datos.
- Orientación a Objetos.
- Tecnología de tarjetas chip.

He aquí la lista de tecnologías que domino, de cada uno de estos tipos, con el enlace al estándar o sitio de referencia de cada una de ellas:

### Tecnologías web.

HTML

<http://www.w3c.org/MarkUp/>

CSS

<http://www.w3c.org/Style/CSS/>

JavaScript

<http://devedge.netscape.com/central/javascript/>

DOM

<http://www.w3.org/DOM/>

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

## XML

<http://www.w3c.org/XML/>

## XSL

<http://www.w3c.org/XSL/>

## RDF/RSS

<http://www.w3c.org/RDF/>

<http://www.w3c.org/RSS/>

## CGI

<http://cgi-spec.golux.com/>

<http://www.w3c.org/CGI/>

[http://httpd.apache.org/docs/mod/mod\\_cgi.html](http://httpd.apache.org/docs/mod/mod_cgi.html)

<http://www.ku.edu/acs/docs/other/forms-intro.html>

## SSI

<http://httpd.apache.org/docs/howto/ssi.html>

[http://httpd.apache.org/docs/mod/mod\\_include.html](http://httpd.apache.org/docs/mod/mod_include.html)

## Servlets

<http://jcp.org/aboutJava/communityprocess/first/jsr154/index3.html>

<http://java.sun.com/products/servlet/>

<http://jakarta.apache.org/tomcat/>

<http://java.sun.com/products/servlet/industry.html>

## Tecnologías de comunicaciones

### HTTP/1.1 (RFC2616, RFC2817)

<ftp://ftp.isi.edu/in-notes/rfc2616.txt>

### HTTP State Management Mechanism (RFC2965)

<ftp://ftp.isi.edu/in-notes/rfc2965.txt>

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

HTTP Authentication: Basic and Digest Access Authentication (RFC2617)

<ftp://ftp.isi.edu/in-notes/rfc2617.txt>

HTTP-NG

<http://www.w3.org/Protocols/HTTP-NG/>

WEBDAV (RFC2518)

<ftp://ftp.isi.edu/in-notes/rfc2518.txt>

<http://www.w3.org/Protocols/>

SSH

<http://www.ietf.org/ids.by.wg/secsh.html>

<http://www.ssh.com>

<http://www.openssh.com>

sockets TCP/IP

<http://theoryx5.uwinnipeg.ca/CPAN/perl/pod/perlipc.html>

<http://theoryx5.uwinnipeg.ca/CPAN/perl/ext/IO/IO/Socket.html>

<http://www.perlfect.com/articles/sockets.shtml>

<http://www.sockets.com>

<http://www.rfc-editor.org/rfc/rfc3493.txt>

protocolos serie (X-Modem, Y-Modem, Z-Modem)

<http://www.ohse.de/uwe/software/lrzs.html>

## Bases de Datos

Diseño de bases de datos

<http://www.agiledata.org/essays/umlDataModelingProfile.html>

SQL

<http://www.wiscorp.com/SQLStandards.html>

<http://database.sarang.net/database/sql/sql.html>

# Enrique Castilla Contreras

Tlf: 661-247020 • E-mail: [ecastillac@yahoo.com](mailto:ecastillac@yahoo.com) • Web: [www.geocities.com/ecastillac](http://www.geocities.com/ecastillac)

---

## Orientación a Objetos

UML

<http://www.omg.org/uml/>

C++

<http://anubis.dkuug.dk/jtc1/sc22/wg21/>

<http://www.parashift.com/c++-faq-lite/>

Java

<http://java.sun.com/docs/>

Perl

<http://www.perl.com>

## Tecnología de tarjetas chip

EURO6000 (CEN/TC224/WG10)

<http://www.europay.com>

<http://www.ecbs.org>

<http://www.diffuse.org/payment.html>

<http://www.cenorm.be/>

EMV

<http://www.emvco.com/>

ISO/IEC 7816-1,2,3

<http://www.iso.org/>

Siemens SLE4442

<http://www.infineon.com>

<http://www.sdlogic.us/downloads/sle4442>