
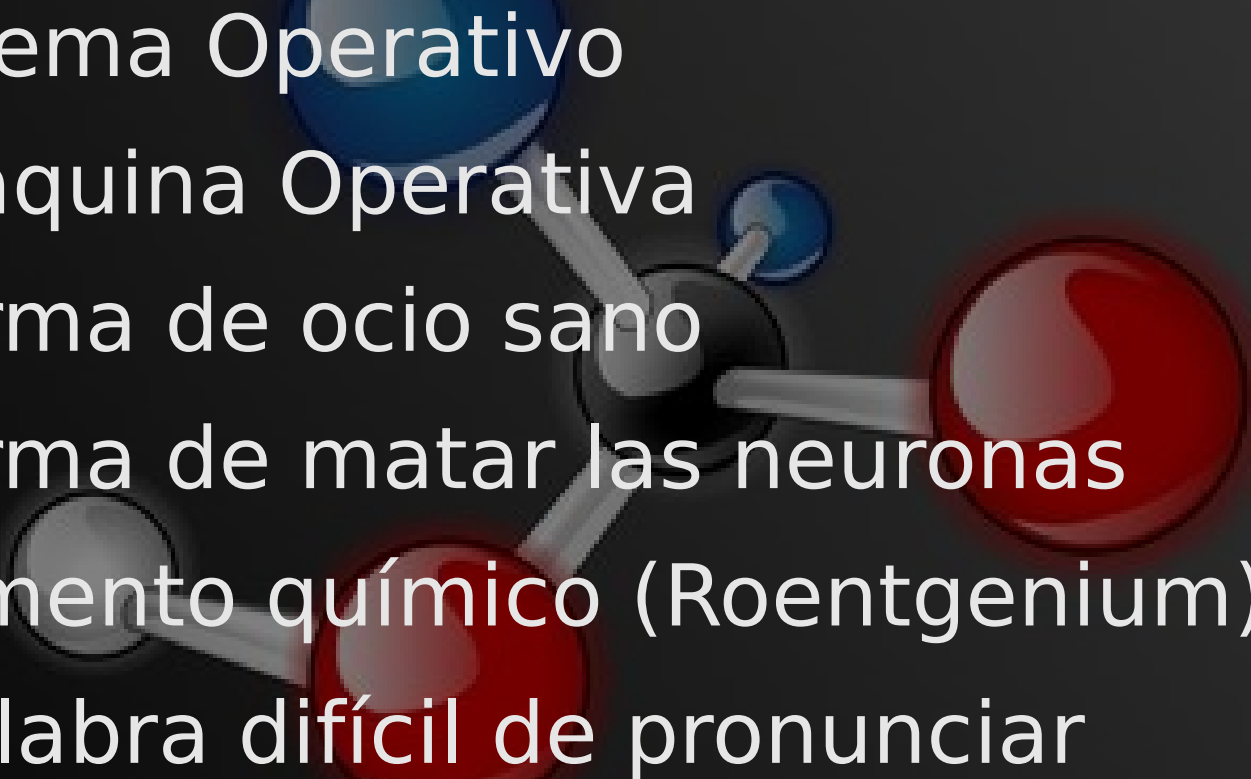


# Unununium



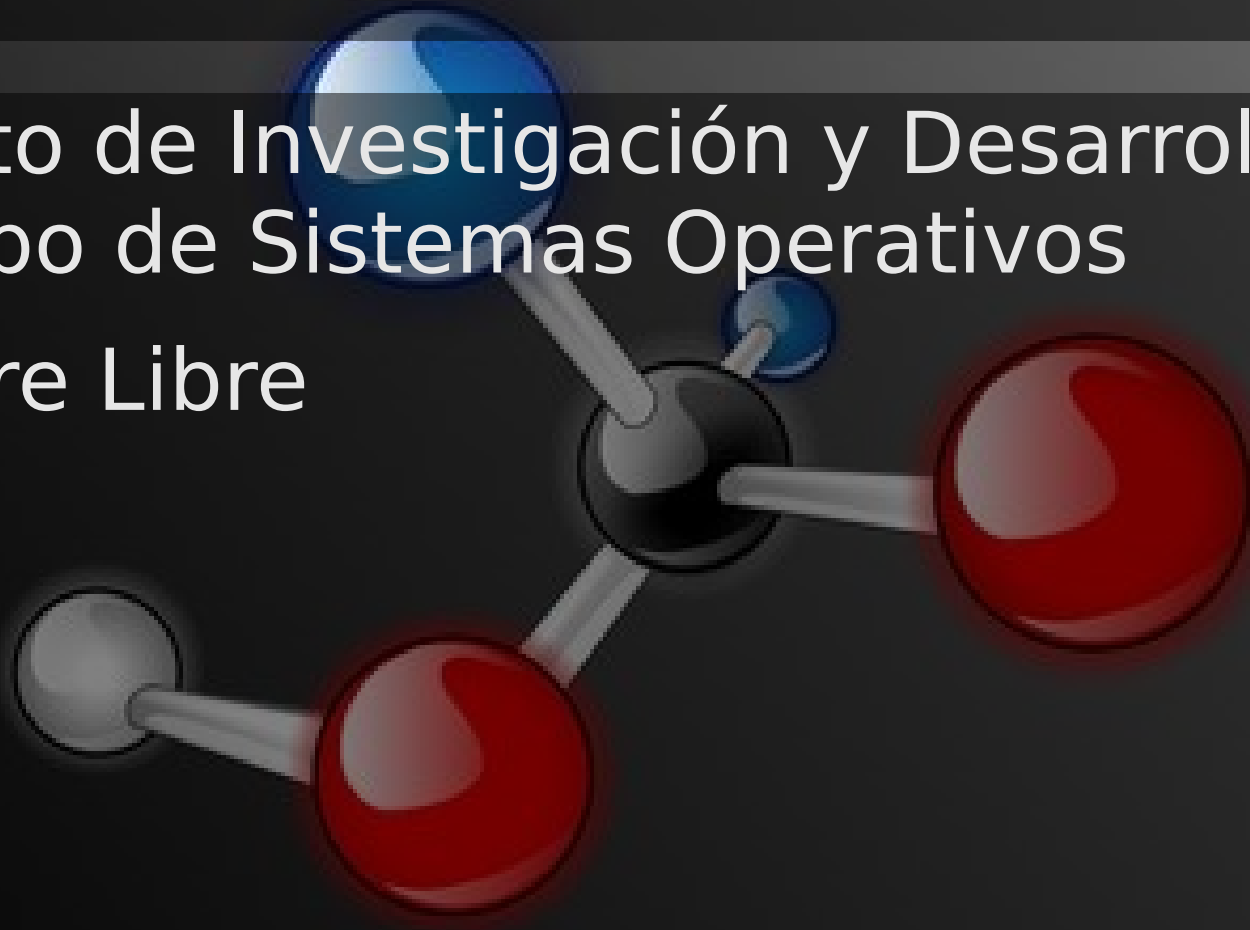
César Yáñez Fernández  
<[cesaryanez@gmail.com](mailto:cesaryanez@gmail.com)>  
Mayo, 2007

# ¿Qué es?

- Un Sistema Operativo
  - Una Máquina Operativa
  - Una forma de ocio sano
  - Una forma de matar las neuronas
  - Un elemento químico (Roentgenium)
  - Una palabra difícil de pronunciar
- 

# ¿Qué es?

- Proyecto de Investigación y Desarrollo en el campo de Sistemas Operativos
- Software Libre



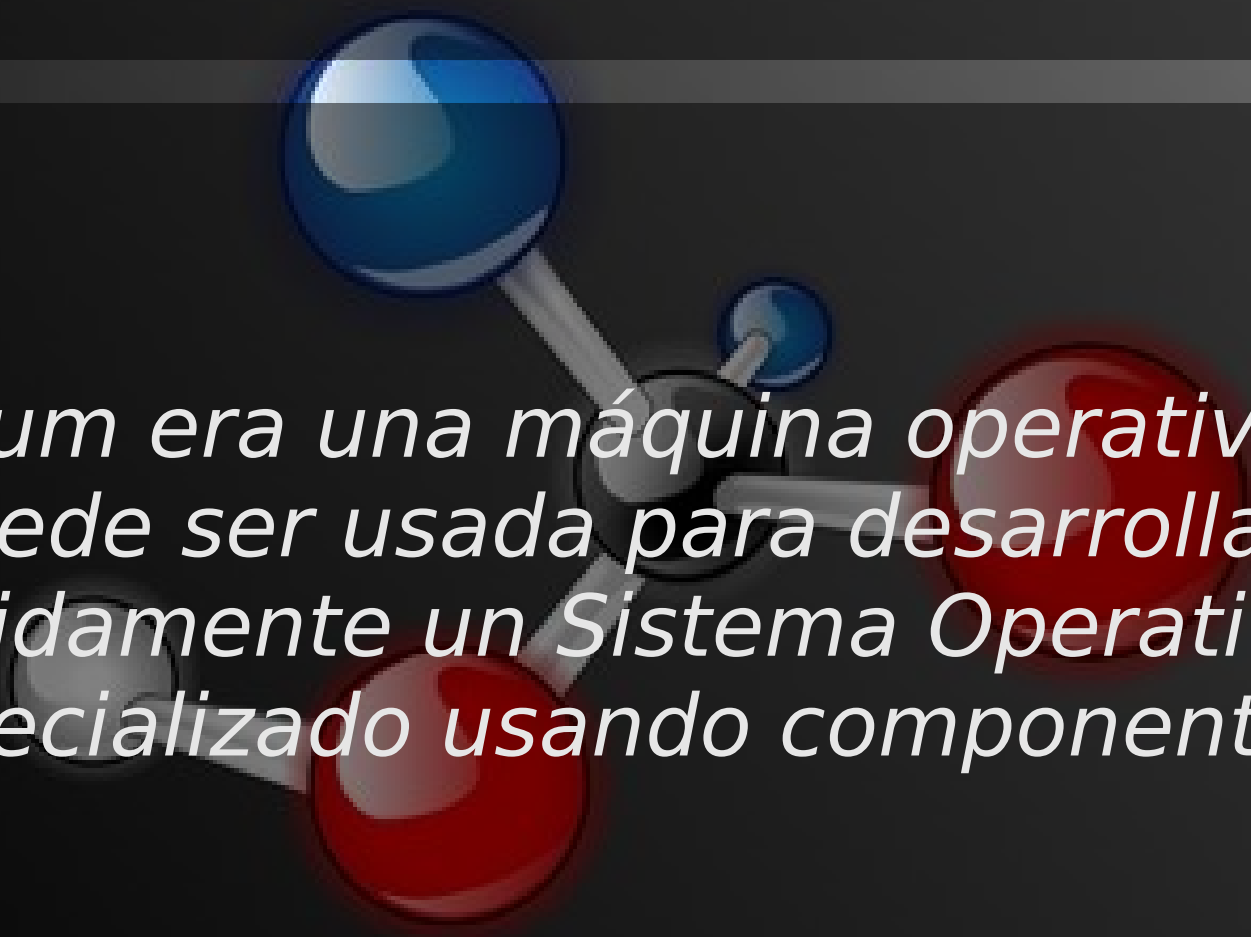
# Cronología

- Octubre de 2000: Nacimiento del proyecto
  - Parte del equipo de *Kernel Coder Team* del proyecto V2\_OS
- 2001: 4<sup>th</sup> Axis Project: Desarrollo del concepto *VOiD*
- 2002: Ununium: Implementación de *SAS*; nace el concepto de *Máquina Operativa*

# Cronología

- 2003: Implementación de *Máquina Virtual, ByteCode* y *JiT Compiler*
- 2004: Instauración de Biblioteca de C (*dietlibc*)
- 2005: Implementación de *POSIX Layer*
- 2006: Adopción de OSKit, instauración de Python como lenguaje de desarrollo
- 2007: Nacimiento de Ich/Yoctu, implementando el concepto de *Programación Concurrente*

# Ununium en el 2002

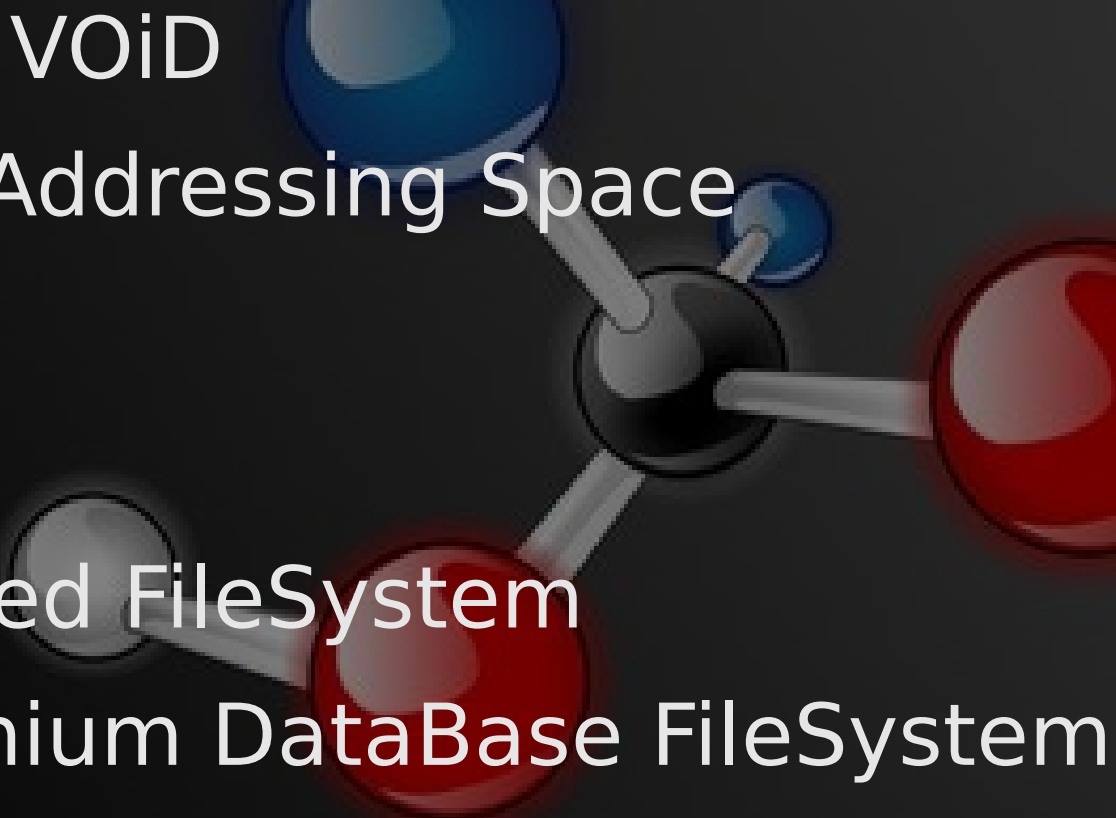


*Ununium era una máquina operativa que puede ser usada para desarrollar rápidamente un Sistema Operativo especializado usando componentes*

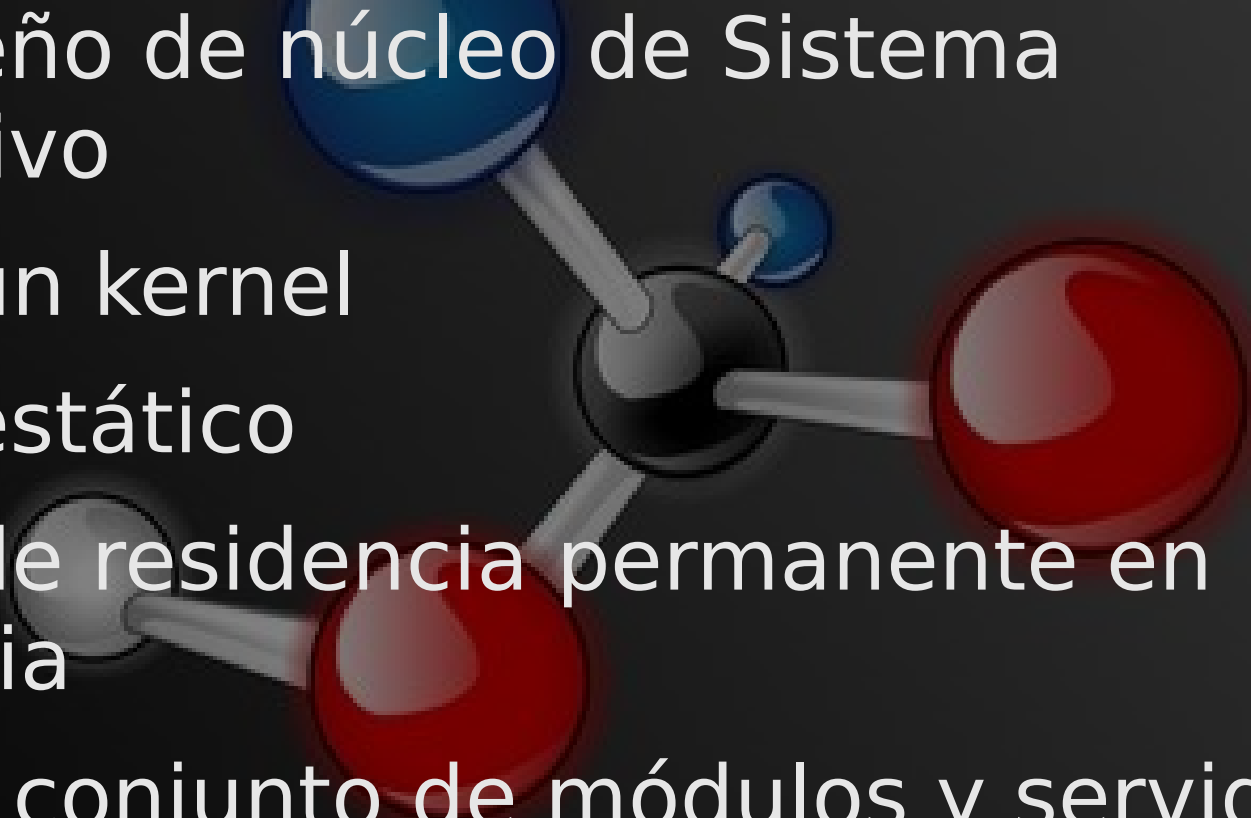
# ¿Qué es una Máquina Operativa?

- Una Máquina Operativa ofrece servicios y un modelo de programación en el cual trabajar.
- Un Sistema Operativo ofrece una interfaz para algunos de esos servicios y también normalmente viene con una serie de herramientas que permite usar el sistema.
- El Sistema Operativo está enfocado al usuario, la Máquina Operativa al software.

# Tecnologías que se han implementado

- Diseño VOiD
  - Single Addressing Space
  - OSKit
  - Python
  - Extended FileSystem
  - Ununium DataBase FileSystem
  - UuuTime (U3T)
- 

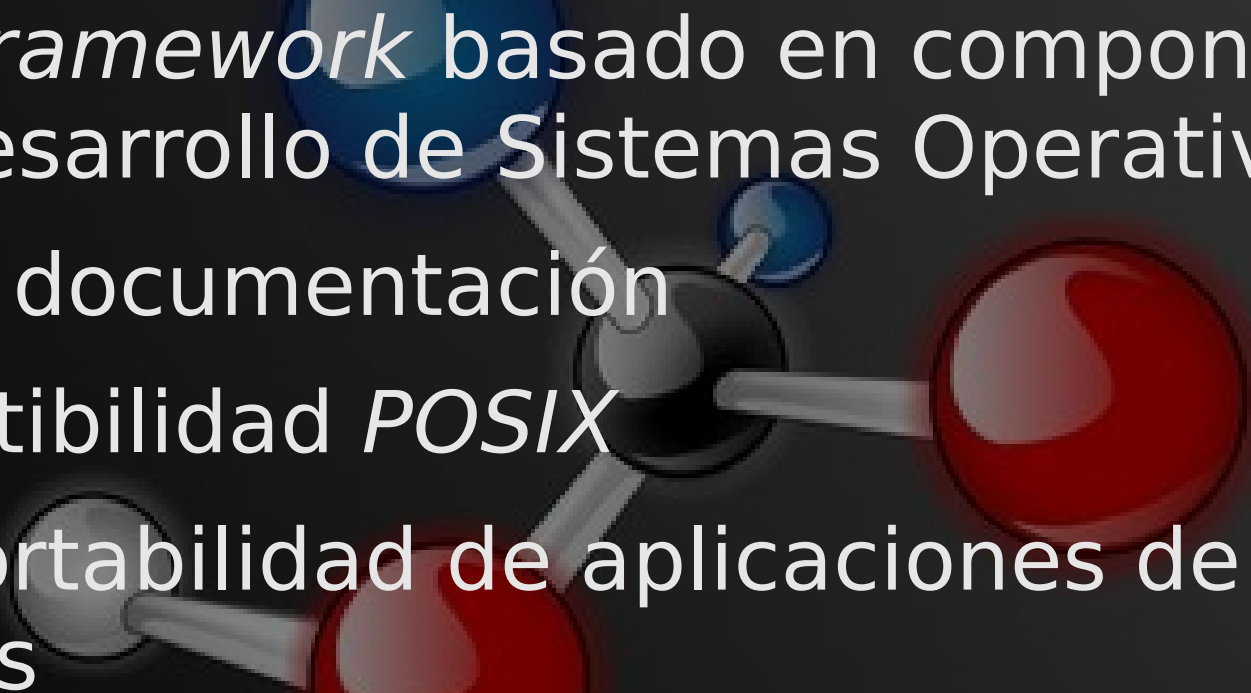
# VOiD

- Un diseño de núcleo de Sistema Operativo
  - No es un kernel
  - No es estático
  - No es de residencia permanente en memoria
  - Son un conjunto de módulos y servidores que proveen un control de funciones de Hardware e Interoperación con Software.
- 

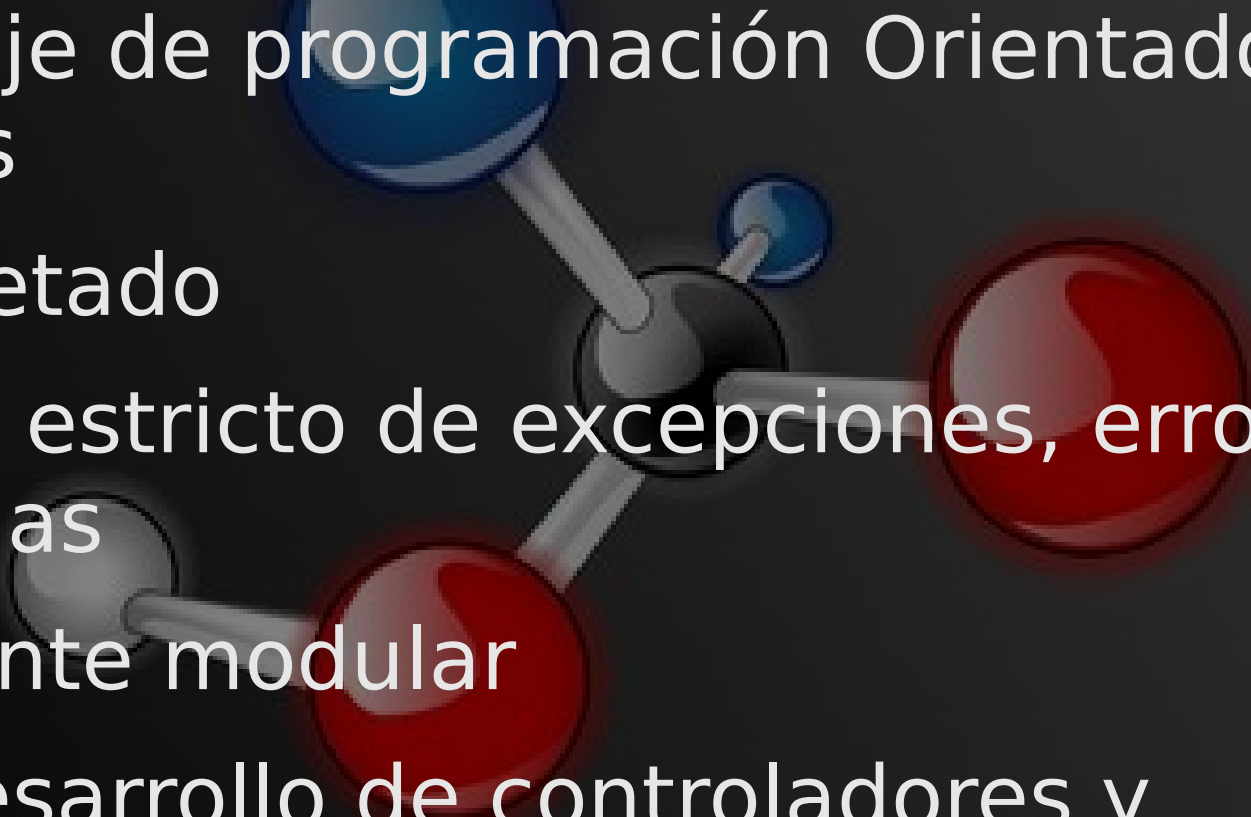
# SAS

- SAS significa *Single Addressing Space*.
- Se define como un modelo para direccionamiento de memoria.
- Permite que varias aplicaciones compartan datos más fácilmente.
- No hay mecanismos de traducción de apuntadores.
- No hay Sistema de Archivos.

# OSKit

- Es un *framework* basado en componentes para desarrollo de Sistemas Operativos
  - Amplia documentación
  - Compatibilidad *POSIX*
  - Fácil portabilidad de aplicaciones de terceros
- 

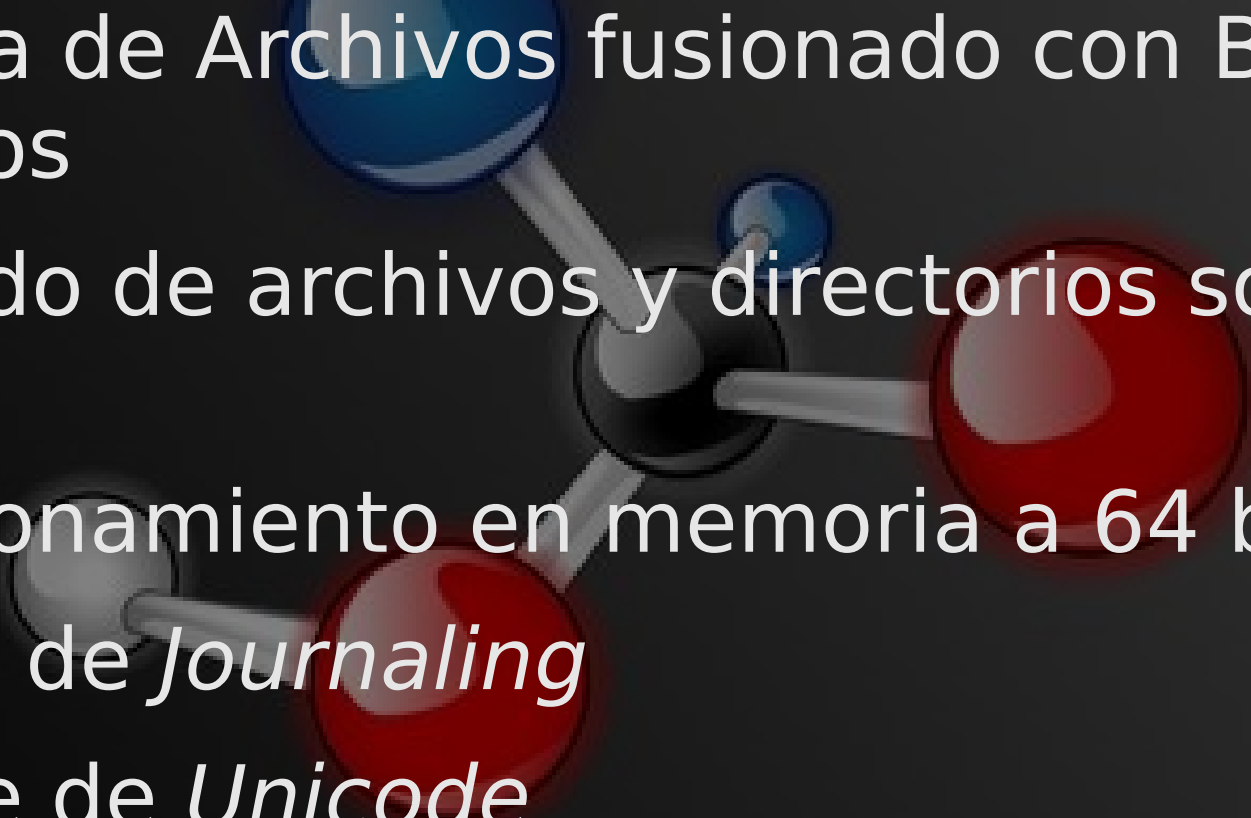
# Python

- Lenguaje de programación Orientado a Objetos
  - Interpretado
  - Control estricto de excepciones, errores y herencias
  - Altamente modular
  - Fácil desarrollo de controladores y aplicaciones
- 

# Extended FileSystem

- El Sistema de Archivos por omisión en Linux (ExtFS)
- Versiones 1, 2, 3, 4; compatibles entre éstas pero con grandes diferencias
- Soporte de Ext2FS en Ununium
- Controlador escrito en 25 líneas de Python
- Altamente portátil

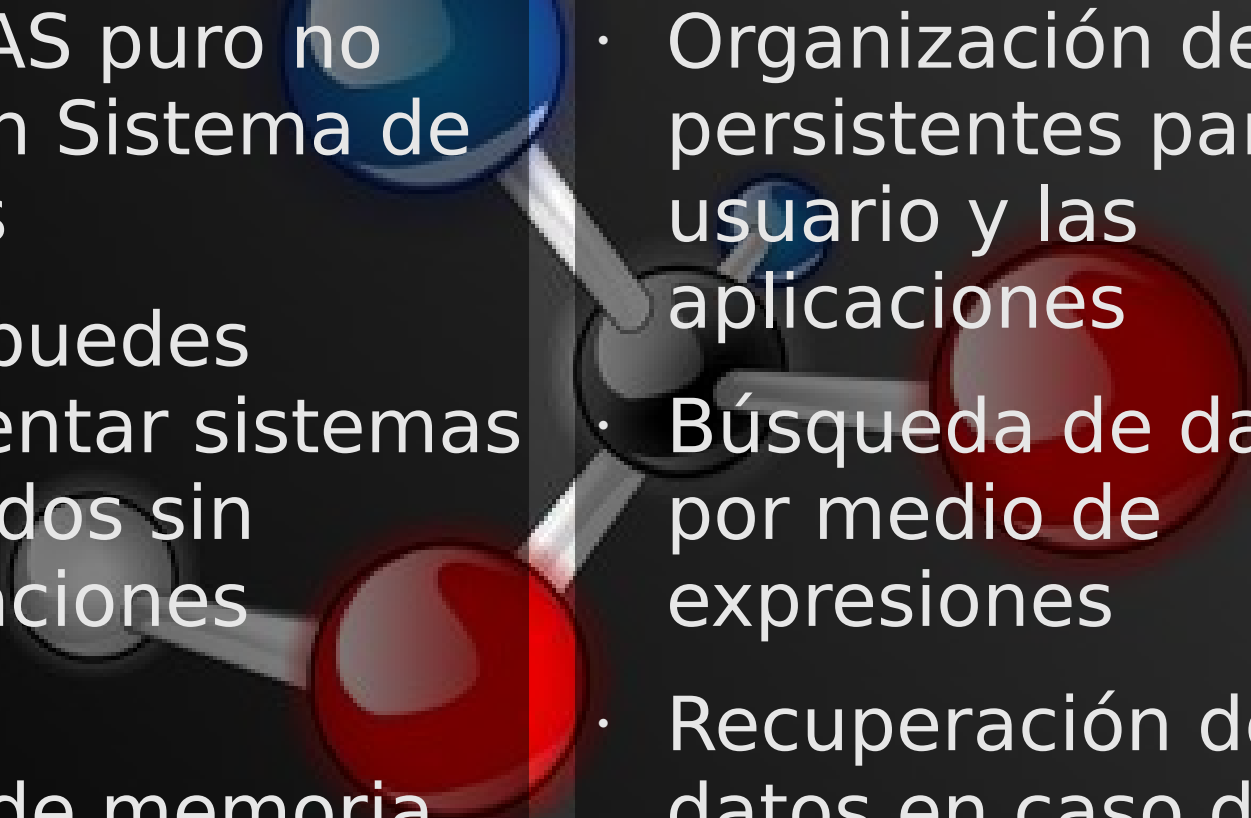
# Ununium DataBase FileSystem

- Sistema de Archivos fusionado con Base de Datos
  - Indexado de archivos y directorios sobre BTrees
  - Direccionamiento en memoria a 64 bits
  - Manejo de *Journaling*
  - Soporte de *Unicode*
- 

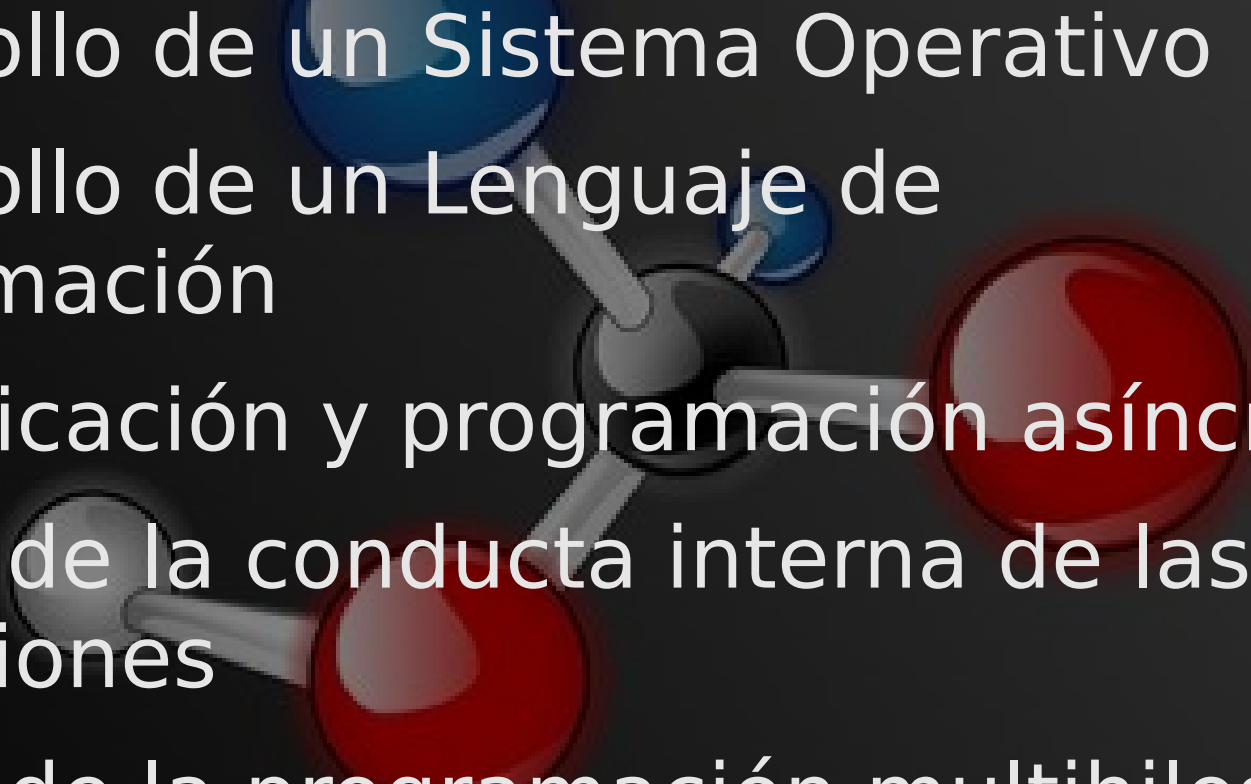
# UuuTime

- Sistema de medición de tiempo en Unununium
- Microsegundos transcurridos a partir del 1 de Enero del 2000, 00:00 UTC (medido con reloj atómico)
- Variable entera en 64 bits sin signo
- Temporización en Tiempo Real Duro

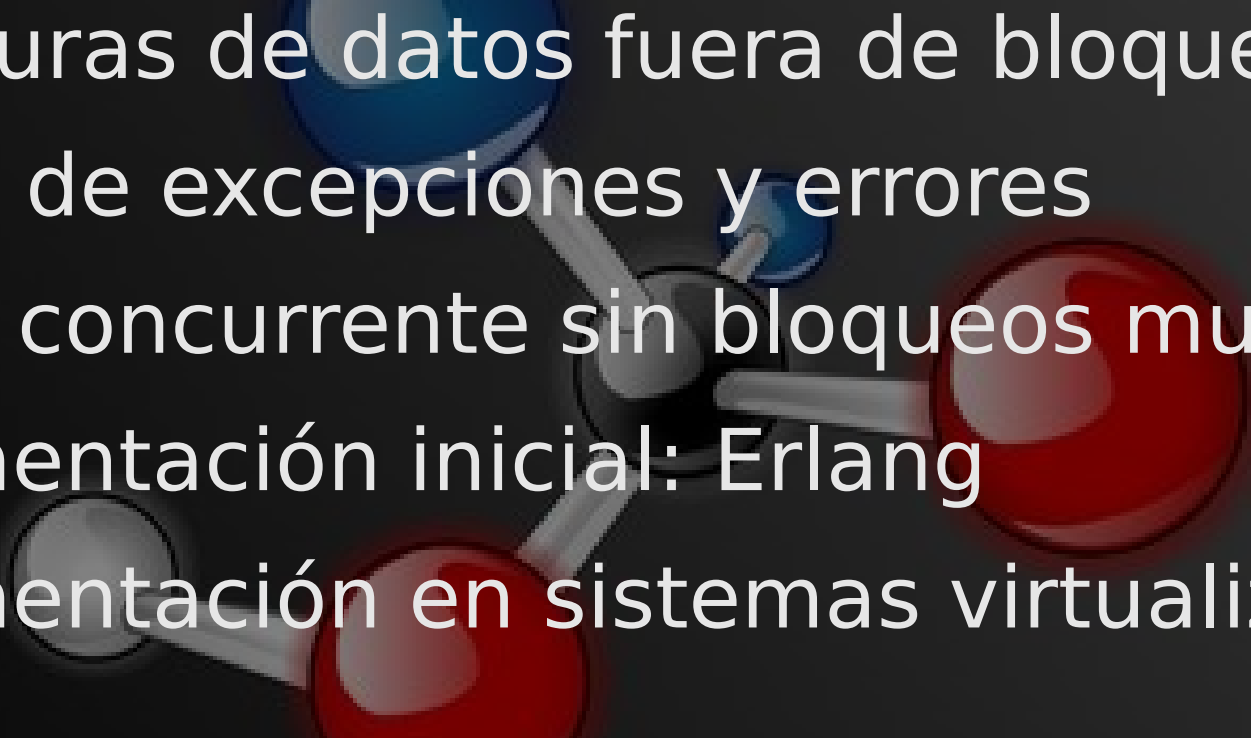
# SAS vs Sistema de Archivos

- 
- En un SAS puro no existe un Sistema de Archivos
  - En SAS puedes implementar sistemas distribuidos sin modificaciones internas
  - Manejo de memoria persistente y memoria volátil como un todo
  - Organización de datos persistentes para el usuario y las aplicaciones
  - Búsqueda de datos por medio de expresiones
  - Recuperación de datos en caso de caída

# El Futuro: Ich/Yoctu

- Desarrollo de un Sistema Operativo
  - Desarrollo de un Lenguaje de Programación
  - Comunicación y programación asíncrona
  - Mejora de la conducta interna de las aplicaciones
  - Mejora de la programación multihilos
  - Control de hilos transparente
- 

# El Futuro: Ich/Yoctu

- Estructuras de datos fuera de bloque
  - Manejo de excepciones y errores
  - Acceso concurrente sin bloqueos mutuos
  - Implementación inicial: Erlang
  - Implementación en sistemas virtualizados (VT-X)
- 

# ¿Preguntas?



<http://hokum.e-ver.com.mx/>