

"huronOS": distribución GNU/Linux orientada a la programación competitiva

Trabajo Terminal No. 2020 - B096

Alumnos: Gómez Rodríguez Enya Quetzalli, Macías Márquez Abraham Omar, González Vélez Bryan Enrique

*Directores: Cortés Galicia Jorge, Saucedo Delgado Rafael Norman
e-mail:gomezreithne@gmail.com, amaciasm1401@alumno.ipn.mx*

Resumen - En este proyecto se desarrollará una distribución GNU/Linux diseñada para live-USB con persistencia de datos, esto se logrará mediante la modificación de una distribución ya existe y el desarrollo de módulos a nivel kernel y a nivel aplicación. Este sistema operativo proveerá a las comunidades de programación competitiva el ambiente y las herramientas necesarias para la realización de una competencia de programación adecuándose a las reglas de las competencias más relevantes del rubro.

Palabras clave - Distribución GNU/Linux, Live-USB, Programación Competitiva, Sistemas Operativos.

1. Introducción

La programación competitiva es un deporte mental que consiste en la resolución de un conjunto de problemas bien-conocidos de Ciencias de la Computación y resolverlos tan rápido como sea posible. [1] Entre las competencias más relevantes se encuentra la "International Collegiate Programming Contest (ICPC)"[2], así como la "International Olympiad in Informatics (IOI)"[3], competencias mundiales en las que participan diversas instituciones educativas alrededor de todo el mundo.

Las entidades que organizan dichos concursos establecen diferentes estratos clasificatorios divididos geográficamente y crean sedes encargadas de la organización local para la participación en el concurso, además, definen una serie de reglas para la realización de los concursos que estipulan el comportamiento permitido por los concursantes, pero también lineamientos, características y recursos que deben proveer las sedes a sus concursantes. Dentro de los recursos que deben ser provistos, se encuentra un entorno de cómputo (ambiente de programación) adecuado para la competencia el cuál, en términos generales, se trata de un sistema operativo GNU/Linux cargado con una serie de herramientas de programación y configuraciones que promuevan el cumplimiento del reglamento para concursantes. [4]

Sin embargo, las diferentes sedes encargadas de la organización de los concursos suelen ser comunidades de programación competitiva de un tamaño pequeño si se comparan con el alumnado dentro de sus instituciones educativas. Motivo por el cual tienen diversas complicaciones:

- Debido a que estas actividades suelen ser consideradas como extracurriculares, tienen una dificultad para el acceso a el equipo de cómputo institucional.
- En caso de que la sede tenga permiso de usar el equipo cómputo institucional, las instituciones pueden tener directivas para la instalación, configuración y uso del equipo de cómputo; por consiguiente, el reemplazo, anexo o modificación del software en el equipo de cómputo se encuentra prohibido.
- Incluso si la sede obtiene permiso para modificar el software de los equipos institucionales, no todas las sedes cuentan con personas capacitadas para la correcta configuración, instalación y adecuación del entorno de programación tal y como fue requisitado por la organización de la competencia, además de tener que devolver los equipos a su estado previo a la modificación.
- Adicional a esto, este proceso debe ser repetido en múltiples ocasiones a lo largo del año dependiendo de la estrategia clasificatoria del concurso hospedado.

Por lo tanto, se propondrá como solución a los problemas anteriormente planteados, el desarrollo de un sistema operativo diseñado para live-USB con persistencia de datos basado en una distribución GNU/Linux ya existente, al cual se le agregarán ciertos módulos (algunos a nivel kernel y otros a nivel aplicación) que proveerán a las comunidades de programación competitiva el ambiente y las herramientas necesarias para la realización de una competencia de programación adecuándose a las reglas de las competencias más relevantes del rubro. Y de esta manera, se quitará, a las comunidades de programación competitiva, la dependencia del software disponible en sus instituciones educativas, además de evitar el proceso de configuración, instalación y adecuación de un entorno tradicional.

1.1. Comunidades de programación competitiva

A pesar de no existir una definición actual sobre lo que es una comunidad de programación competitiva, la Comunidad de Programación ICPC de la ITESO definió a una comunidad ICPC en la región México como:

“Una comunidad académica en la que participen alumnos, académicos, egresados y personal de la industria, de una o más instituciones, alrededor de concursos de programación, campamentos, entrenamientos, estrategias de preparación en programación incluidas en currículos de programas de estudio en instituciones, etc.; relacionados con el concurso ICPC.”[2]

Se propone una definición, basada en lo anterior, de lo que es una comunidad de programación competitiva como:

Una comunidad académica en la que participen alumnos, académicos, egresados y personal de la industria de una institución educativa, alrededor de concursos de programación, campamentos, entrenamientos, estrategias de preparación en programación y otras actividades relacionadas a la programación competitiva.

Normalmente una sede que decide hospedar una competencia de un concurso de programación es una comunidad de programación competitiva que ha tomado una cierta relevancia en su área geográfica y ha obtenido los permisos necesarios de su institución educativa para hacer uso del equipo institucional, no solo para la propia comunidad sino para otros individuos o comunidades externas que estén inscritas en el concurso hospedado. Por este motivo, el público objetivo son las comunidades de programación competitiva y no las sedes de un concurso de programación, ya que el proyecto podría suplir no solo la necesidad específica de un concurso, si no de todo un conjunto de comunidades dedicadas a la programación competitiva y todas las actividades que ello implica.

1.2. Sistemas similares

Se hizo una búsqueda de sistemas similares que se han desarrollado, entre ellas se encuentran:

SOFTWARE	CARACTERÍSTICAS	PRECIO
Distribución Linux “IXCHEL” <i>Trabajo de Tesis de la Escuela Superior de Ingeniería Mecánica y Eléctrica</i>	Se trata de una implementación de una distribución GNU/Linux orientada al perfil estudiantil de la carrera ICE de la ESIME. La distribución busca cubrir las necesidades estudiantiles en cuanto a software y aplicaciones, basándose, en las unidades de aprendizaje del mapa curricular de dicha carrera. (ESIME,2017)[5].	No disponible
Maratona-Linux: um ambiente para a Maratona de Programacao <i>Proyecto de la Universidad Tecnológica Federal de Paraná</i>	Se trata de un conjunto de aplicaciones para Debian desarrolladas para atender los requerimientos de la región ICPC Brasil organizado por la Sociedad Brasileña de Computación. Maratona-Linux es el entorno predeterminado en todos los sitios y por todos los equipos en Brasil desde 2016. (UTFPR, 2019) [6].	Gratuito

SOFTWARE	CARACTERÍSTICAS	PRECIO
NomadBSD <i>Sistema operativo basado en FreeBSD</i>	NomadBSD, es un sistema operativo de propósito general basado en FreeBSD, orientado a live-USB con persistencia de datos y con características de auto-detección de hardware. (The NomadBSD Team, 2020) [7]	Gratuito

Cuadro 1: Resumen de productos similares

2. Objetivo

Modificar una distribución GNU/Linux mediante la incorporación de un conjunto de módulos a nivel kernel y a nivel aplicación que provea a las comunidades hispanohablantes de programación competitiva el ambiente adecuado y las herramientas necesarias para la realización de una competencia de programación.

2.1. Objetivos específicos

1. Desarrollar y anexar un módulo de persistencia de datos para unidades de almacenamiento USB que le permita al sistema operativo funcionar como sistema live-USB sin tener pérdida de datos.
2. Desarrollar y anexar un módulo de sincronización y aplicación de directivas que le permita al sistema operativo obtener las configuraciones de comportamiento establecidas por las comunidades y, además, aplicarlas en el equipo.
3. Desarrollar y anexar un módulo de control de acceso a la red que le permita al sistema operativo aplicar las directivas configuradas.
4. Desarrollar y anexar un módulo de control de archivos y software que le permita al sistema operativo otorgar permisos a los usuarios sobre la lectura, escritura y ejecución de los archivos y, además, eliminarlos de acuerdo a las directivas establecidas.
5. Instalar el software que le permita a los módulos controlar el acceso a las herramientas de programación comúnmente requeridas por las competencias más relevantes.
6. Definir un entorno de escritorio para el sistema operativo que le permita adecuarse a diferentes capacidades de hardware.

3. Justificación

Dadas las diferentes dificultades que enfrentan diversas comunidades de programación competitiva a lo largo del mundo para crear entornos adecuados para la realización de sus competencias, se propone una distribución GNU/Linux orientada a la programación competitiva que permitirá a las comunidades una mayor accesibilidad a un entorno apto para una competencia de programación, evitar la dependencia de un sistema instalado en un disco duro, tener mayor flexibilidad para adaptarse a distintas condiciones de hardware y configurar la distribución de una manera que les beneficie.

Actualmente existen tres alternativas para configurar un entorno adecuado para las competencias: configuración manual, imagen de la mundial de ICPC y el conjunto de aplicaciones Maratona-Linux, sin embargo ninguna de estas alternativas es una solución integral que considere las diferentes situaciones que pasan las comunidades de programación competitiva, por lo que huronOS, al ser una distribución GNU/Linux, resultaría en un producto robusto e integral capaz de atender las necesidades que ningún otro trabajo ha podido solucionar hasta la fecha.

Para el caso de este proyecto se agregarán algunos módulos de nivel kernel a una distribución GNU/Linux ya existente, cuya implementación debe realizarse con sumo cuidado para evitar tener un núcleo inestable, además, agregar módulos a nivel aplicación que permitan implementar el resto de características deseadas, las cuales serán

necesarias para cumplir con la totalidad de la solución; sin embargo, dada la magnitud de la misma, los tiempos disponibles y la cantidad de miembros en el presente equipo de trabajo, el proyecto se delimitará tentativamente a un único módulo de nivel kernel y tres posibles módulos de nivel aplicación, además, se limitará al idioma español por lo que principalmente se atenderá sólo a las comunidades hispanohablantes, bajo la premisa de que la solución total pueda ser completada en trabajos posteriores.

4. Productos o resultados esperados

Al concluir este trabajo terminal, se modificará una distribución GNU/Linux para que provea a las comunidades de programación competitiva un ambiente apto para la realización de una competencia de programación, y contenga herramientas adecuadas para el seguimiento de las reglas del concurso; esto se logrará mediante el diseño, arquitectura, desarrollo e implementación de los siguientes módulos (ordenados por prioridad):

1. Módulo de persistencia de datos

Un sistema live-USB puede arrancar el sistema operativo directo desde una unidad flash, o copiando el sistema operativo a la RAM. En la primera opción se utiliza la unidad misma como medio de almacenamiento, evitando pérdidas de datos pero limitando la velocidad de trabajo; en la segunda opción se tiene una gran velocidad de trabajo, pero al ser la RAM el medio de almacenamiento los datos son volátiles. Para ganar ambos beneficios, se utilizará la segunda opción y se diseñará un módulo de persistencia de datos que escribe sobre la unidad flash los datos que se quieran almacenar para futuros arranques del sistema. Este módulo se diseñará para adaptarse a las necesidades a solucionar en este sistema operativo, proporcionando persistencia para archivos creados por el usuario final, pero también configuraciones, software, y directivas para el uso del sistema operativo.

2. Módulo de sincronización y aplicación de directivas

Dado que uno de los objetivos específicos de éste proyecto es evitar la frecuente preparación de un ambiente de programación para la competencias, el módulo de sincronización y aplicación de directivas, permitirá al sistema operativo acceder a las directivas establecidas por la comunidad de programación competitiva que configuró la distribución (hora de internet, fondos de pantalla, horarios de concursos, modalidades de funcionamiento, modalidades de control de red, permisos de ejecución, etc.), y se aplicarán mediante el uso de otros módulos de control, de forma que las directivas establecidas se mantengan siempre actualizadas sin necesitar reconfigurar el sistema operativo.

3. Módulo de control de acceso a la red.

Dado que la mayoría de las competencias importantes de programación competitiva únicamente permiten material impreso (libros, referencias, códigos, apuntes o teoría), este módulo se encargará de restringir el acceso a recursos digitales (redes sociales, foros de ayuda, mensajería instantánea, repositorios, wikis, buscadores y almacenamiento en la nube, etc.) no solo de internet, sino también de la red de área local, y además, permitirá el acceso al servidor que hospede el juez en línea que controlará el concurso configurado, así como a los servicios de sincronización de directivas.

4. Módulo de control de archivos y software

Un sistema live-USB suele ser ejecutado esporádicamente a diferencia de sistemas operativos de uso personal, por lo que existen casos en los que el usuario final no eliminará ningún archivo programado durante alguna competencia, y el equipo encargado de la administración de la competencia deberá asegurarse de eliminar manualmente dichos archivos residuales. Esto genera una mayor carga de trabajo y produce vulnerabilidades para el acceso al material digital en futuros concursos.

Por ello, este módulo se encargará de otorgar o denegar al usuario final, permisos de lectura, escritura y/o ejecución sobre archivos binarios de software y archivos creados por el usuario; además de tener la capacidad de *limpiar* los directorios del sistema operativo para dar la apariencia de que se trata de un sistema recién instalado. Los indicadores de qué acciones tomar sobre los archivos del sistema, estarán dados por las directivas provistas por el módulo de sincronización y aplicación de directivas.

Se determinará la forma de implementación de estos módulos (nivel kernel o nivel aplicación) así como las tecnologías a utilizar para su implementación, durante la fase de diseño y arquitectura del presente trabajo terminal.

Como trabajo adicional, pero, sin compromiso de realizar su análisis, diseño, arquitectura, desarrollo e implementación, y, con total libertad de preservarlo para futuros desarrollos, se espera poder desarrollar un instalador para el sistema operativo que facilite su configuración e instalación en las unidades USB.

También, a lo largo del desarrollo de este trabajo terminal, se obtendrá el manual de usuario y el manual técnico.

5. Metodología

Debido a las características de este proyecto, se decidió utilizar la metodología de Desarrollo de Software de Código Abierto (Open Source Development Model)[8]. Esta metodología generalmente es usada en el desarrollo de características de distribuciones Linux.

Modelo de Desarrollo de Código Abierto
The open source development model

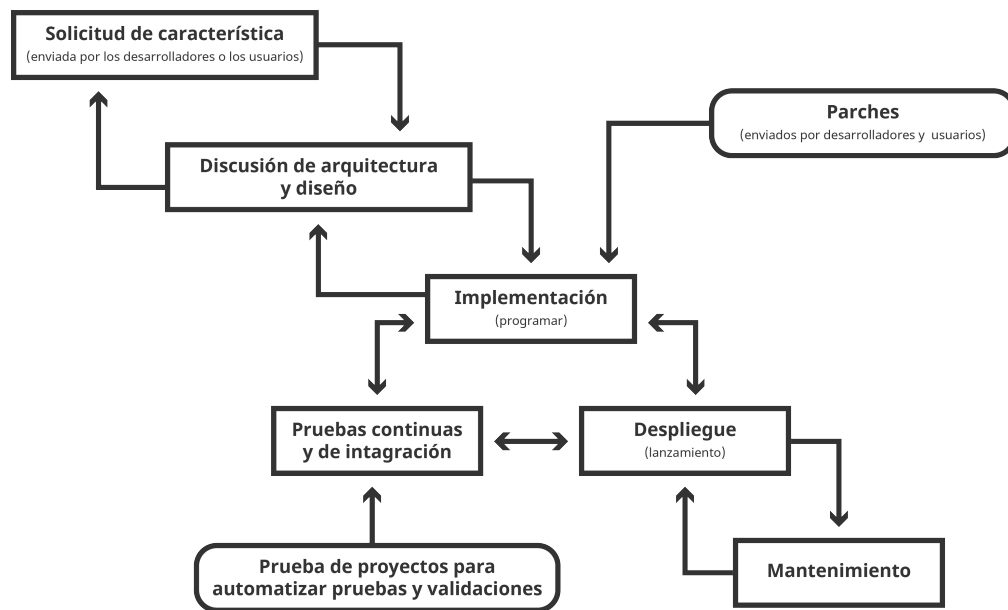


Figura 1: Metodología de Desarrollo de Software Libre

El ciclo de desarrollo de software en esta metodología se muestra en la figura 1. Empieza cuando un usuario o un desarrollador envía la solicitud de una nueva característica a una lista de espera. Los colaboradores del proyecto revisan cada solicitud para determinar si debe ser aceptada, replanteada o rechazada. Cuando es aceptada empieza la discusión sobre el diseño de la nueva característica, la cual se debe realizar por medio de un canal donde todas las personas involucradas en el proyecto, incluyendo a los usuarios finales, puedan participar. Se hace gran énfasis en que la fase de implementación del código sea colaborativa y que facilite la retroalimentación entre colaboradores. Cuando el código escrito ya es funcional, puede ser enviado a revisión para determinar si es lanzado. Es recomendable que el incremento de cada envío sea lo más pequeño posible con el objetivo de facilitar la revisión, detectar errores y recibir retroalimentación de los usuarios finales de forma constante. [8]

Como en este proyecto se realizará la modificación de una distribución GNU/Linux, esta metodología se ajusta perfectamente a nuestras necesidades. Además, enfatiza la constante comunicación con los usuarios en diferentes etapas del proyecto, lo cual se considera importante para los objetivos de este trabajo.

6. Cronogramas

CRONOGRAMA Nombre de la alumna: Gómez Rodríguez Enya Quetzalli

TT No.: 2020-B096

Título del TT: "huronOS : distribución GNU/Linux orientada a la programación competitiva"

Actividad	FEB	MAR	ABR	MAY	JUN	JUL	FEB	MAR	ABR	MAY	JUN
Elaboración de marco teórico											
Obtención de requisitos funcionales y no funcionales											
Investigación del estado del arte											
Descripción de características											
Discusión de diseño y arquitectura											
Investigación y aprendizaje sobre tecnologías a utilizar											
Evaluación de TT1											
Implementación de las características											
Despliegue											
Pruebas											
Generación de manual de usuario											
Generación de manual técnico											
Evaluación de TT2											

CRONOGRAMA Nombre del alumno: González Vélez Bryan Enrique

TT No.: 2020-B096

Título del TT: "huronOS : distribución GNU/Linux orientada a la programación competitiva"

Actividad	FEB	MAR	ABR	MAY	JUN	JUL	FEB	MAR	ABR	MAY	JUN
Elaboración de marco teórico											
Obtención de requisitos funcionales y no funcionales											
Investigación del estado del arte											
Descripción de características											
Discusión de diseño y arquitectura											
Investigación y aprendizaje sobre tecnologías a utilizar											
Evaluación de TT1											
Implementación de las características											
Despliegue											
Pruebas											
Generación de manual de usuario											
Generación de manual técnico											
Evaluación de TT2											

CRONOGRAMA Nombre del alumno: Macías Márquez Abraham Omar

TT No.: 2020-B096

Título del TT: "huronOS : distribución GNU/Linux orientada a la programación competitiva"

Actividad	FEB	MAR	ABR	MAY	JUN	JUL	FEB	MAR	ABR	MAY	JUN
Elaboración de marco teórico											
Obtención de requisitos funcionales y no funcionales											
Investigación del estado del arte											
Descripción de características											
Discusión de diseño y arquitectura											
Investigación y aprendizaje sobre tecnologías a utilizar											
Evaluación de TT1											
Implementación de las características											
Despliegue											
Pruebas											
Generación de manual de usuario											
Generación de manual técnico											
Evaluación de TT2											

7. Referencias

- [1]Halim, S., Halim, F., Rodríguez, R. M. (2019). *Programación competitiva: Manual para concursantes del ICPC y la IOI* (1.a ed., Vol. 1). OJBooks. <https://cpbook.net/>
- [2]ICPC. (s. f.). *ICPC - About ICPC*. The ICPC International Collegiate Programming Contest. Recuperado 29 de marzo de 2021, de <https://icpc.global/regionals/abouticpc>
- [3]Opmanis, M. *History of the International Olympiad in Informatics*. International Olympiad in Informatics. Recuperado 29 de marzo de 2021, de <http://history.ioinformatics.org/history.shtml>
- [4]ICPC. (s. f.). *ICPC - Programming Environment*. The ICPC International Collegiate Programming Contest. Recuperado 7 de noviembre de 2020, de <https://icpc.global/worldfinals/programming-environment>
- [5]I.A. Molina, J.C. Camacho, “Distribución LINUX IXCHEL”. Tesis de Ingeniería, ESIME, CDMX, 2016.
- [6]Morais, Wall Ribas, Bruno. (2019). Maratona-Linux: um ambiente para a Maratona de Programação.
- [7]The NomadBSD Team, *NomadBSD - About*. [online] Disponible en: <<https://nomadbsd.org>>[Recupero 16 de marzo de 2021].
- [8]Haddad, I. Warner, B. (2011). *Understanding the Open Source Development Model*. The Linux Foundation, Noviembre 2011. Recuperado 18 de marzo de 2020, de <https://www.ibrahimatlinux.com/uploads/6/3/9/7/6397792/00.pdf>.

8. Alumnos y Directores

CARÁCTER: Confidencial
FUNDAMENTO LEGAL: Artículo 11 Fracc. V y Artículos
108, 113 y 117 de la Ley Federal de Transparencia y Acceso
a la Información Pública.
PARTES CONFIDENCIALES: Número de boleta y teléfono.

Gómez Rodríguez Enya Quetzalli.- Alumna de la carrera de Ingeniería en Sistemas Computacionales en la Escuela Superior de Cómputo del Instituto Politécnico Nacional (ESCOM-IPN). Boleta: 2014030529, Tel. +52 1 55 6611 0725, email: gomezreithne@gmail.com

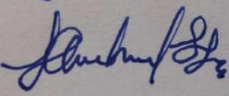
González Vélez Bryan Enrique.- Alumno de la carrera de Ingeniería en Sistemas Computacionales en la Escuela Superior de Cómputo del Instituto Politécnico Nacional (ESCOM-IPN). Boleta: 2017630691, Tel. +52 1 55 2181 4395, email: begvyazg@gmail.com


Macías Márquez Abraham Omar.- Alumno de la carrera de Ingeniería en Sistemas Computacionales en la Escuela Superior de Cómputo del Instituto Politécnico Nacional (ESCOM-IPN), Boleta: 2015302235, Tel. +52 1 55 3657 3492, email: amaciasm1401@alumno.ipn.mx

Cortés Galicia Jorge.- Director. Dr. en Ciencias de la Computación del CIC-IPN en 2014, M. en C. en Computación del CIC-IPN en 2001, Ing. en Electrónica de la UAM en 1998, Profesor de ESCOM/IPN (Depto de ISC) desde 2002, Áreas de Interés: Sistemas Operativos, Sistemas Distribuidos, Inteligencia Artificial. Teléfono: 57296000 Ext. 52039, e-mail: jcortesg@ipn.mx.

Saucedo Delgado Rafael Norman.- M. en C. Saucedo Delgado Rafael Norman.- Ing. en Sistemas Computacionales, Egresado del Instituto Politécnico Nacional, M. en C. en Ciencias de la Computación, por el IIMAS-UNAM. Profesor de la ESCOM en el Departamento de Ciencias e Ingeniería de la Computación. Áreas de Interés: Gráficos por Computadora y Realidad Virtual, Teléfono: 57296000 Ext. 52022, email. rsaucedo@ipn.mx


Enyo Quetzalli Gómez Rodríguez

Acuse de recibido

Abraham Omar Macías Márquez



González Vélez
Bryan Enrique
Acuso de recibido

 **Jorge Cortes Galicia**
to me ▾
Buenas noches,

Estoy de acuerdo.

Saludos.
Jorge Cortés Galicia

De: Daniel Gonzalez <danielgonznun@gmail.com>
Enviado: lunes, 9 de noviembre de 2020 19:21
Para: Jorge Cortes Galicia <jcortesg@ipn.mx>
Cc: gomezreithne@gmail.com <gomezreithne@gmail.com>; Abraham Omar Macias Marquez <amaciasm1401@alumno.ipn.mx>
Asunto: Protocolo y Firma: huronOS

 **Rafael Norman Saucedo Delgado**
to me ▾
Estoy de acuerdo

M. En C. Rafael Norman Saucedo Delgado

From: Daniel Gonzalez <danielgonznun@gmail.com>
Sent: Monday, November 9, 2020 21:19
To: Rafael Norman Saucedo Delgado <rsaucedo@ipn.mx>
Subject: Firma de Protocolo: huronOS

...