

INSTITUTO TECNOLÓGICO DE LA PAZ
INGENIERÍA EN SISTEMAS COMPUTACIONALES

**Herramienta web para la integración de sistemas de
predicción de regiones de unión al complejo mayor de
histocompatibilidad clase II**

REPORTE FINAL DE RESIDENCIA

PRESENTA:
JUAN LUIS SALAZAR MENDOZA

ASESOR INTERNO:
DR. MARCO ANTONIO CASTRO LIERA

ASESOR EXTERNO:
DR. CARLOS ELIUD ANGULO VALADEZ



LA PAZ, BAJA CALIFORNIA SUR, MÉXICO, JUNIO 2014.



AGRADECIMIENTOS

Le doy gracias a mi madre María Elena, por su ejemplo y empeño para que yo pudiese lograr mis metas. A Enrique que siempre ha estado presente y pendiente de cómo van mis estudios.

A mis hermanos Daniel, Luisa y Rogelio, quienes a pesar de nuestras diferencias me han apoyado con su paciencia y entendimiento.

A mis maestros por su tiempo, esfuerzo y consejos, dentro y fuera de las aulas.

A Dr. Marco Antonio Castro Liera por asesorarme en las cuestiones técnicas del proyecto, por ayudarme cuando me daba de topes con el programa, por su confianza y sobre todo por sus enseñanzas.

A la IBQ. Perla Carlos García por enseñarme a utilizar muchas herramientas inmunoinformáticas y por mostrarme a través de su tesis la trascendencia de este proyecto.

A Dr. Carlos Eliud Angulo Valadez del CIBNOR por invitarme a participar en su proyecto de investigación, además por su paciencia y entusiasmo al ilustrarme con tanto detalle el sistema inmunológico.

RESUMEN

En este documento se presenta toda la información sobre el proyecto final de residencias, dividido en dos capítulos. El capítulo 1 es el informe final de residencias. El capítulo 2 son los documentos administrativos del proyecto.

El informe final de residencias es la parte más extensa del documento y se divide en trece secciones. En la sección 1.1 “Introducción” se presenta el propósito del trabajo con algunos antecedentes para ubicar al lector en el desarrollo del proyecto. En la sección 1.2 “Datos generales de la empresa” se muestran los datos del centro de investigación con el que se participa. En la sección 1.3 “Justificación del proyecto” se explica las razones por las cuales fue necesario el desarrollo del proyecto. En la sección 1.4 “Objetivos generales y específicos” se presentan los objetivos. En la sección 1.5 “Características del área en que participó” se explica el área en la que se desarrolló el proyecto. En la sección 1.6 “Alcances y limitaciones” se expone los alcances y limitaciones así como la situación legal del producto. En la sección 1.7 “Fundamento o Marco Teórico” están las bases teóricas que fueron necesarias para el desarrollo del proyecto. En la sección 1.8 “Procedimientos y descripción de las actividades realizadas” se detalla el desarrollo del proyecto. En la sección 1.9 “Evaluación o impacto económico, social o tecnológico” se explica el impacto del proyecto en el área de aplicación. En la sección 1.10 “Resultados Obtenidos” se exponen los resultados. En la sección 1.11 “Conclusiones y recomendaciones” las conclusiones y recomendaciones para trabajo futuro. En la sección 1.12 “Bibliografía” las fuentes de información. En la sección 1.13 “Anexos” los elementos anexados.

ÍNDICE

1 Informe Final de Residencia Profesional	6
1.1 Introducción	6
1.2 Datos generales de la empresa	7
1.2.1 Nombre de la empresa	7
1.2.2 Domicilio de la empresa	7
1.2.3 Giro	7
1.2.4 Organigrama	7
1.2.5 Breve descripción de la empresa	8
1.3 Justificación del proyecto.....	9
1.4 Objetivos generales y específicos.....	10
1.4.1 Objetivo general	10
1.4.2 Objetivos específicos	10
1.5 Características del área en que participó.....	11
1.6 Alcances y limitaciones.....	12
1.7 Fundamento o Marco Teórico.....	13
1.7.1 Conceptos biológicos	13
1.7.2 Bioinformática.....	18
1.7.3 Inmunología Computacional.....	19
1.7.4 Vacunología inversa.....	19
1.7.5 Arquitectura cliente - servidor.....	20
1.7.6 HTML	21
1.7.7 PHP	22

1.7.8 Javascript	23
1.7.9 Expresiones regulares.....	24
1.7.10 NetMHCpan.....	25
1.7.11 NetMHCIIPan.....	25
1.7.12 Cello	25
1.7.13 BepiPred	26
1.7.14 Formato FASTA	26
1.8 Procedimiento y descripción de las actividades realizadas	27
1.8.1 Selección de las herramientas inmunoinformáticas a integrar	27
1.8.2 Integración de interfaces	27
1.8.3 Comunicación con los servidores.....	34
1.8.4 Parseo de respuestas	41
1.8.5 Filtrado por localización subcelular	63
1.8.6 Creación de reportes.....	65
1.9 Evaluación o impacto económico, social o tecnológico	68
1.10 Resultados Obtenidos.....	69
1.11 Conclusiones y recomendaciones	70
1.12 Bibliografía	71
1.13 Anexos: planos, gráficas, prototipos y diagramas.....	72
1.13.1 Anexo 1. Pantalla principal de Cello v.2.5	72
1.13.2 Anexo 2. Pantalla principal de NetMHCpan v.2.8.....	73
1.13.3 Anexo 3. Pantalla principal de NetMHCIIPan v3.0.....	74
1.13.4 Anexo 4. Pantalla principal de BepiPred v.1.0.....	75
1.13.5 Anexo 5. Diagrama de caso de uso.	76

1.13.6 Anexo 6. Diagrama de clases.....	79
1.13.7 Anexo 7. Clases parse de cada servidor.....	80
1.13.8 Anexo 8. Clase Fasta.....	93
1.13.9 Anexo 9. Clase Reports.	94
2 Documentos administrativos	100
2. 1 Carta de presentación y convenio con la empresa	100
2.2 Evaluación interna	103
2.3 Evaluación externa	104

1 Informe Final de Residencia Profesional

1.1 Introducción

El proyecto “Herramienta web para la integración de sistemas de predicción de regiones de unión al complejo mayor de histocompatibilidad clase II” fue desarrollado como una propuesta de solución a una problemática del Centro de Investigaciones Biológicas del Noroeste S.C.

La problemática surge de la necesidad de una herramienta que permita automatizar la creación de reportes a partir de la interacción con ciertas aplicaciones web que realizan un análisis exhaustivo de secuencias de aminoácidos. Este análisis consiste en la predicción de regiones de unión al complejo mayor de histocompatibilidad clase II (MHCII), complementado con otros sistemas que dan soporte experimental a estos resultados, tales como: localización subcelular, predicción de unión al complejo mayor de histocompatibilidad clase I (MHCI) y Localización de epítopos de células B dentro de las secuencias presentadas.

Como solución se desarrolló una aplicación web que integra cuatro herramientas existentes y de uso habitual en la actualidad para estos fines. La integración se logró en una sola interfaz que permite redireccionar las secuencias de aminoácidos y los parámetros de entrada para cada aplicación web. Los resultados de cada herramienta se obtienen y se generan reportes automáticamente con la interpretación de los resultados.

1.2 Datos generales de la empresa

1.2.1 Nombre de la empresa

Centro de Investigaciones Biológicas del Noroeste, S.C.

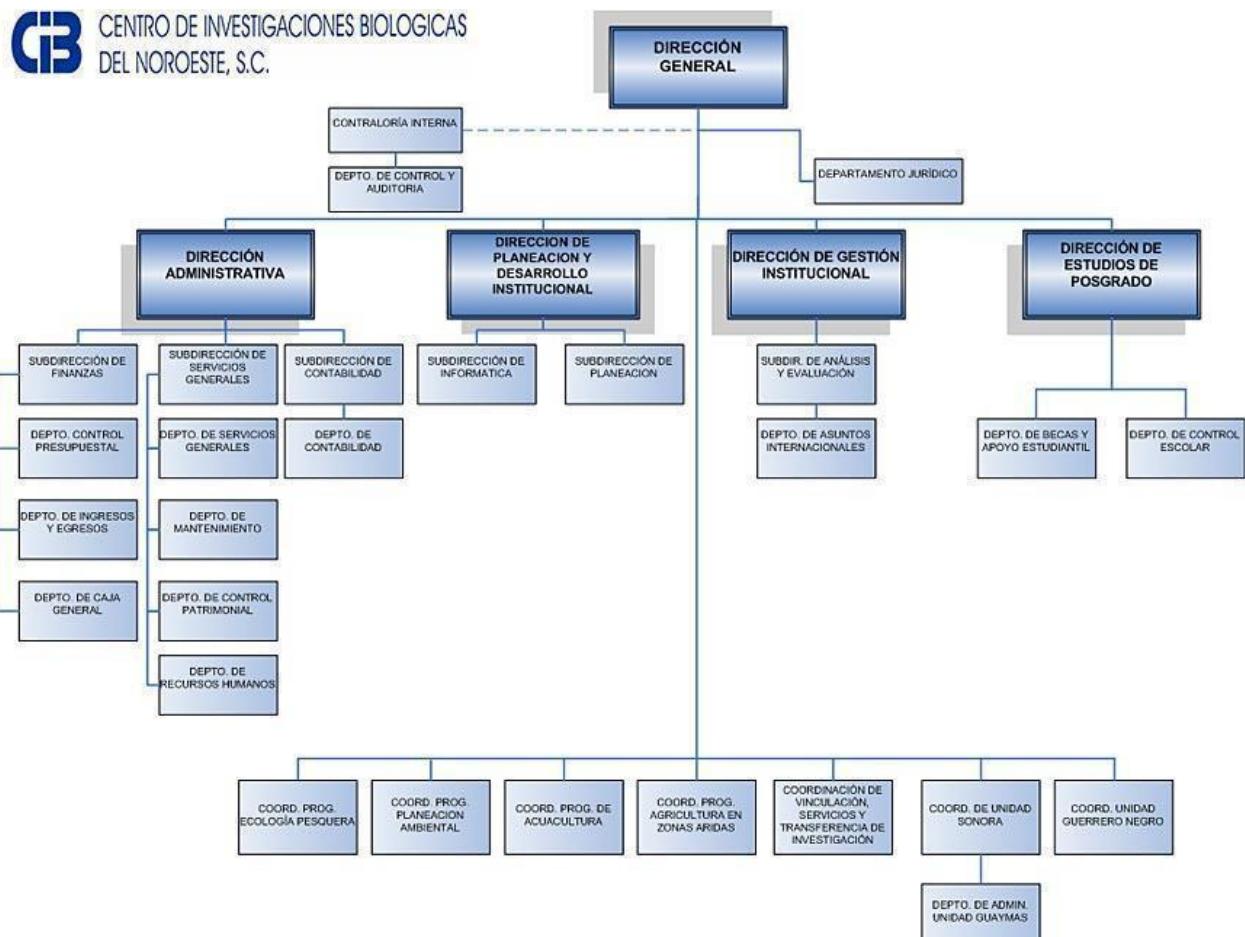
1.2.2 Domicilio de la empresa

Mar bermejo número 195, col. Playa Palo de Santa Rita, La Paz, B.C.S. México; C.P. 23096.

1.2.3 Giro

Centro de investigación CONACYT.

1.2.4 Organigrama



1.2.5 Breve descripción de la empresa

El Centro de Investigaciones Biológicas del Noroeste, S.C., es un centro de investigación perteneciente al Sistema de Centros Públicos CONACYT, cuya misión es coadyuvar al bienestar de la sociedad mediante la realización de investigación científica, innovación tecnológica y formación de recursos humanos, en el manejo sustentable de los recursos naturales.

Uno de los programas activos de este Centro de Investigaciones es el de “Agricultura en Zonas Áridas” donde se insertan las líneas de generación y aplicación del conocimiento de Biotecnología Animal, Inmunología y Vacunología. Entre los objetivos que se persigue dentro de las líneas antes mencionadas se establece el desarrollo de vacunas para el ganado de la región.

1.3 Justificación del proyecto

En la actualidad existen varias aplicaciones web que permiten realizar predicciones de unión péptido – molécula MHCII, pero, dado que cada herramienta utiliza una metodología diferente, los resultados obtenidos con cada una son distintos, además cada aplicación utiliza un formato particular para presentar los resultados, lo cual dificulta su comparación y análisis.

Aunado a lo anterior, no resulta práctico contrastar resultados entre sistemas pues la comparación se hace manualmente resultado por resultado, y por lo tanto el proceso es lento.

Es por esto que la elaboración de una herramienta que tenga la capacidad de análisis de los sistemas actuales de predicción y que logre consolidar los resultados de manera automática permitirá ahorrar tiempo y esfuerzo a los investigadores, con lo que se logrará aumentar la velocidad de análisis, experimentación y por consiguiente la productividad.

1.4 Objetivos generales y específicos

1.4.1 Objetivo general

Desarrollar una aplicación web integradora que permita predecir la ubicación subcelular de proteínas de patógenos y los péptidos de unión a células T y B, que utilice la información generada por diferentes sistemas y metodologías de predicción para generar reportes consolidados y ponderados de los resultados.

1.4.2 Objetivos específicos

- Seleccionar las herramientas inmunoinformáticas relevantes para la predicción de unión de epítopos de proteínas al MHC I, MHCII, Células B y la predicción de la localización subcelular de las proteínas.
- Comprender los mecanismos computacionales para obtener la información de dichas herramientas.
- Construir una sola interfaz web de forma local y que redireccione a los servidores seleccionados.
- Diseñar los reportes que se mostrarán como resultado de la interpretación y análisis de los resultados obtenidos.
- Integrar todos los módulos y presentar el producto ante los usuarios finales.

1.5 Características del área en que participó

El presente trabajo se enmarca en la disciplina de inmunoinformática. Para el desarrollo del proyecto se utilizaron herramientas tecnológicas disponibles en la web. Este proyecto forma parte de las líneas de investigación del Programa de Agricultura en Zonas Áridas del Centro de Investigaciones Biológicas del Noroeste, SC. Adicionalmente, el presente trabajo forma parte del proyecto de Ciencia Básica SEP-CONACYT Identificación y Evaluación de Vacunas Comestibles en Alfalfa para Prevenir la Paratuberculosis (*Mycobacterium avium* subsp. *paratuberculosis*) de rumiantes domésticos de México (No. 151818). Las competencias profesionales desarrolladas durante la ejecución de este proyecto son capacidad para usar herramientas inmunoinformáticas y bioinformáticas.

1.6 Alcances y limitaciones

El sistema hace uso de servidores web externos (Cello, NetMHCI, NetMHCII, BepiPred), es por ello que se necesita forzosamente el acceso a Internet, además, el buen funcionamiento de nuestro sistema está limitado a la disponibilidad de los servidores antes mencionados.

El usuario tiene a su disposición la elección de los servidores a utilizar, así como la edición de parámetros predeterminados de cada uno.

El sistema local puede utilizarse de manera indefinida pero con las limitaciones particulares de cada servidor externo, estas limitaciones pueden variar según sus criterios.

El mantenimiento de la aplicación debe estar a cargo de alguien que conozca el código a profundidad, el propósito de esto es evitar la reescritura de código y alentar a la reusabilidad en futuras actualizaciones.

El software cuenta con licencia GPL versión 2.0 por lo cual se garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Bajo esta filosofía, la licencia GPL garantiza a los usuarios los derechos y libertades reunidos en definición de software libre e implementa el concepto de copyleft para asegurar que el software está protegido cada vez que el trabajo es distribuido, modificado o ampliado.

.

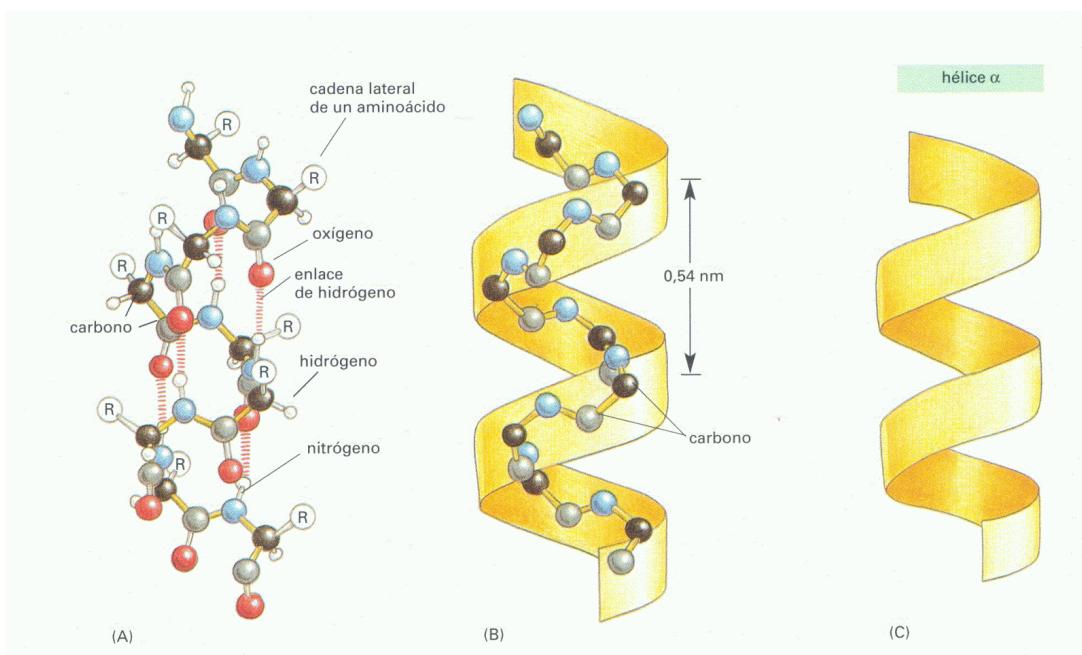
1.7 Fundamento o Marco Teórico

Es importante mencionar las herramientas tecnológicas y conceptos que conforman el fundamento necesario para el desarrollo del proyecto.

1.7.1 Conceptos biológicos

Proteína

Las proteínas son moléculas formadas por cadenas lineales de aminoácidos. Son necesarias para la vida, sobre todo por su función plástica (constituyen el 80% del protoplasma deshidratado de toda célula), pero también por sus funciones biorreguladoras (forman parte de las enzimas) y de defensa (los anticuerpos son proteínas).

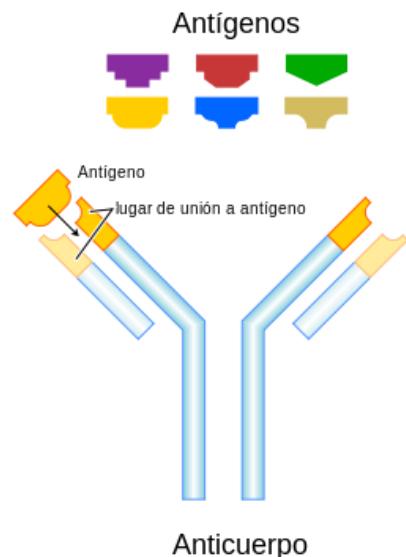


Prácticamente todos los procesos biológicos dependen de la presencia o la actividad de este tipo de moléculas. Bastan algunos ejemplos para dar idea de la variedad y trascendencia de las funciones que desempeñan. Son proteínas: Casi todas las enzimas, catalizadores de reacciones químicas en organismos vivientes, la hemoglobina y otras moléculas con funciones de transporte en la sangre, los

anticuerpos, encargados de acciones de defensa natural contra infecciones o agentes patógenos, los receptores de las células, a los cuales se fijan moléculas capaces de desencadenar una respuesta determinada, la actina y la miosina, responsables finales del acortamiento del músculo durante la contracción, entre otros.

Antígeno

Un antígeno es una sustancia que desencadena la formación de anticuerpos y puede causar una respuesta inmunitaria. La definición moderna abarca todas las sustancias que pueden ser reconocidas por el sistema inmune adaptativo, bien sean propias o ajena. Los antígenos son usualmente proteínas o polisacáridos. Esto incluye partes de bacterias (cápsula, pared celular, flagelos, fimbrias, y toxinas), de virus y otros microorganismos



Anticuerpos

Los anticuerpos (también conocidos como inmunoglobulinas, abreviado Ig) son glicoproteínas del tipo gamma globulina. Pueden encontrarse de forma soluble en la sangre u otros fluidos corporales de los vertebrados, disponiendo de una forma idéntica que actúa como receptor de los linfocitos B y son empleados por el sistema inmunitario para identificar y neutralizar elementos extraños tales como bacterias, virus o parásitos.

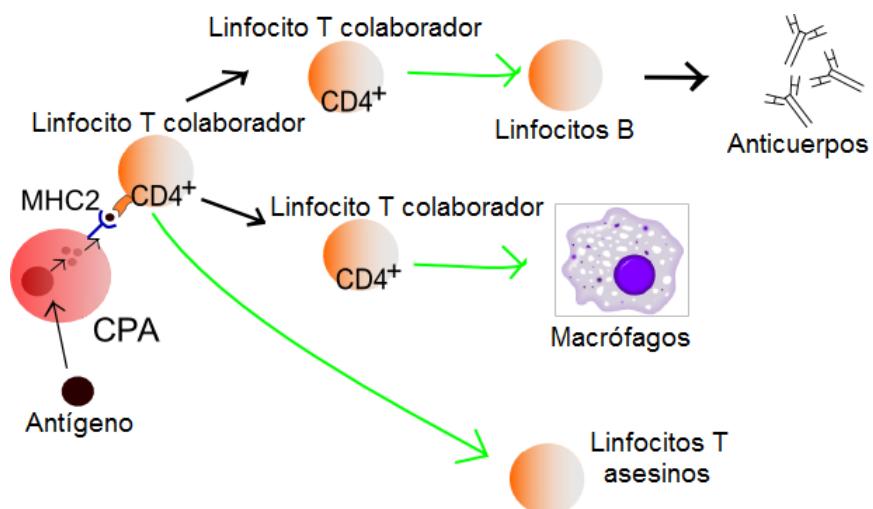
Aminoácidos

La unión de varios aminoácidos da lugar a cadenas llamadas péptidos o polipéptidos, que se denominan proteínas cuando la cadena polipeptídica supera una cierta longitud (entre 50 y 100 residuos aminoácidos, dependiendo de los autores) o la masa molecular total supera las 5000 una y, especialmente, cuando tienen una estructura tridimensional estable definida.

Nº	AMINOÁCIDO	ABREVIATURA	SÍMBOLO
1	Alanina	Ala	A
2	Cisteína	Cys	C
3	Ácido aspártico	Asp	D
4	Ácido glutámico	Glu	E
5	Fenilalanina	Phe	F
6	Glicina	Gly	G
7	Histidina	His	H
8	Isoleucina	Ile	I
9	Lisina	Lys	K
10	Leucina	Leu	L
11	Metionina	Met	M
12	Asparagina	Asn	N
13	Pirrolisina	Pyl	O
14	Prolina	Pro	P
15	Glutamina	Gln	Q
16	Arginina	Arg	R
17	Serina	Ser	S
18	Treonina	Thr	T
19	Selenocisteína	Sec	U
20	Valina	Val	V
21	Triptófano	Trp	W
22	Tirosina	Tyr	Y

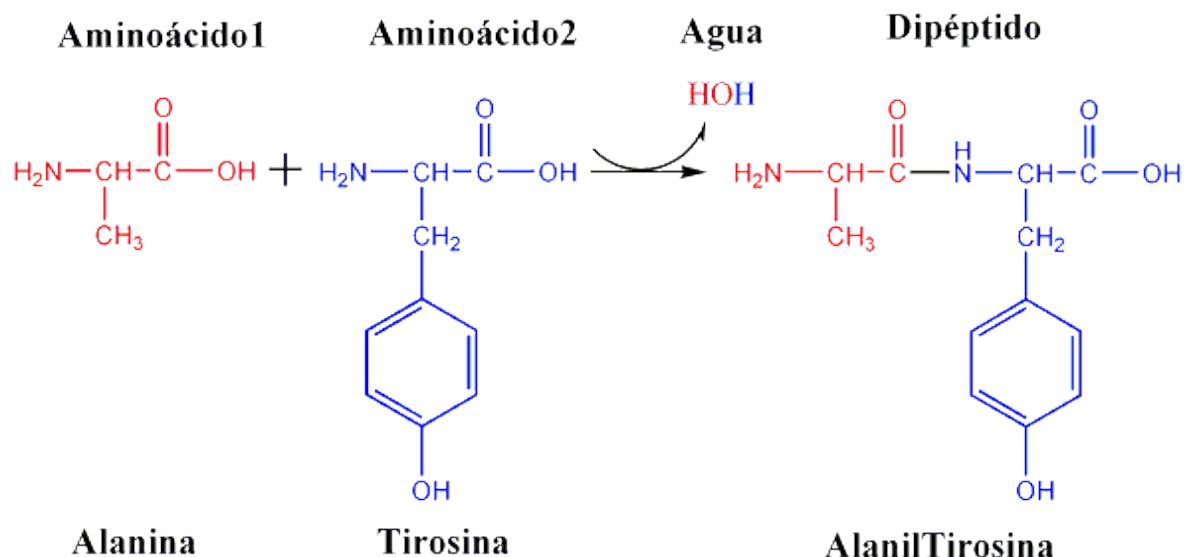
Linfocitos B

Los linfocitos B son los leucocitos de los cuales depende la inmunidad mediada por anticuerpos con actividad específica de fijación de antígenos. Las células B, que constituyen un 5 a 15% del total de linfocitos, dan origen a las células plasmáticas que producen anticuerpos.



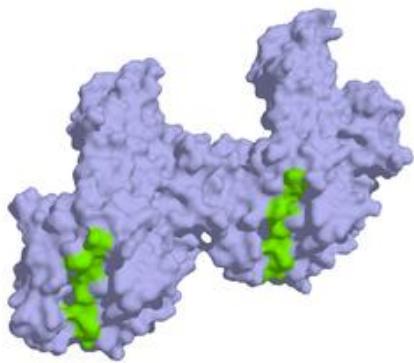
Péptido

La unión de un bajo número de aminoácidos da lugar a un péptido, y si el número es alto, a una proteína, aunque los límites entre ambos no están definidos.



Epítopo

Los anticuerpos, bien sea libres o fijados a la matriz extracelular, se unen a moléculas antigenicas en una superficie de unión, para así formar complejos antígeno-anticuerpo. Esas superficies de unión sobre las macromoléculas constan de complejas secuencias específicas llamadas determinantes antigenicos o epítulos. La mayoría de los epítulos reconocidos por anticuerpos o células B se pueden pensar como relieves de superficies tridimensionales de una molécula antígeno; estos relieves encajan con precisión y así se unen a anticuerpos. La parte de un anticuerpo que reconoce el epítopo se llama paratopo.



Complejo mayor de histocompatibilidad

El complejo mayor de histocompatibilidad (CMH o MHC, acrónimo para el inglés major histocompatibility complex), o complejo principal de histocompatibilidad, es una familia de genes ubicados en el brazo corto del cromosoma 6 cuyos productos están implicados en la presentación de antígenos a los linfocitos T y en la diferenciación de lo propio y lo ajeno en el sistema inmunitario.

En humanos, los genes MHC conforman el denominado sistema HLA (por human leukocyte antigen), porque estas proteínas se descubrieron como antígenos en los leucocitos, que podían detectarse con anticuerpos. Los genes MHC son fundamentales en la defensa inmunológica del organismo frente a los patógenos, y por otro lado, constituyen la principal barrera al trasplante de órganos y de células madre.

La región del brazo corto del cromosoma 6 que contiene los genes del MHC posee la información de:

- ciertas glucoproteínas de la membrana plasmática involucradas en los mecanismos de presentación y procesamiento de antígenos a los linfocitos T: se agrupan en los genes de clase II (que codifican las proteínas MHC-II) y los genes de clase I (que codifican las proteínas MHC-I)
- así como citocinas y proteínas del sistema del complemento, importantes en la respuesta inmunológica, pero que no tienen nada que ver con los genes del MHC; estos genes se agrupan en la clase III.

Ambos tipos de moléculas participan en la respuesta inmunitaria, que permite la identificación de las moléculas propias y de las extrañas (invasoras), para eliminar estas últimas mediante diferentes mecanismos.

Alelos

Un alelo o aleoide es cada una de las formas alternativas que puede tener un mismo gen que se diferencian en su secuencia y que se puede manifestar en modificaciones concretas de la función de ese gen. (Producen variaciones en características heredadas como, por ejemplo, el color de ojos o el grupo sanguíneo).

Por alelo debe entenderse el valor de dominio que se otorga a un gen cuando rivaliza contra otro gen por la ocupación de posición final en los cromosomas durante la separación que se produce durante la meiosis celular.

1.7.2 Bioinformática

La bioinformática, según una de sus definiciones más sencillas, es la aplicación de tecnología de computadores a la gestión y análisis de datos biológicos. Los términos bioinformática, biología computacional y, en ocasiones, biocomputación, utilizados en muchas situaciones como sinónimos, hacen referencia a campos de estudios interdisciplinares muy vinculados que requieren el uso o el desarrollo de diferentes técnicas estudiadas universitariamente en la Ingeniería Informática como ciencia aplicada de la disciplina informática. Entre estas pueden destacarse las siguientes: matemática aplicada, estadística, ciencias de la computación, inteligencia artificial, química y bioquímica con las que el Ingeniero Informático soluciona problemas al analizar datos, o simular sistemas o mecanismos, todos ellos de índole biológica, y usualmente (pero no de forma exclusiva) en el nivel molecular. El núcleo principal de estas técnicas se encuentra en la utilización de recursos computacionales para solucionar o investigar problemas sobre escalas de tal magnitud que sobrepasan el discernimiento humano. La investigación en biología computacional se solapa a menudo con la biología de sistemas.

Los principales esfuerzos de investigación en estos campos incluyen el alineamiento de secuencias, la predicción de genes, montaje del genoma, alineamiento estructural de proteínas, predicción de estructura de proteínas, predicción de la expresión génica, interacciones proteína-proteína, y modelado de la evolución.

1.7.3 Inmunología Computacional

Inmunología Computacional comenzó hace más de 90 años con el modelado teórico de la epidemiología de la malaria. En ese momento, se hizo hincapié en el uso de las matemáticas para guiar el estudio de la transmisión de la enfermedad. Desde entonces, el campo se ha ampliado para cubrir todos los aspectos de los procesos del sistema inmunológico y enfermedades.

Actualmente el principal objetivo del campo es convertir los datos inmunológicos en problemas de cálculo, resolver estos problemas usando matemáticas y computacionales enfoques y luego convertir estos resultados en interpretaciones inmunológicamente significativas.

1.7.4 Vacunología inversa

Vacunología inversa es una mejora en la vacunología , que emplea la bioinformática, la primera vez por Rino Rappuoli y utilizó por primera vez contra el serogrupo B del meningococo. Desde entonces, se ha utilizado en varias otras vacunas bacterianas.

La idea básica detrás de la vacunología inversa es que todo un patógeno genoma pueden ser examinados mediante enfoques bioinformáticos para encontrar genes. Algunos de los rasgos de los genes son controlados para que puedan indicar antigenicidad incluyen genes que codifican para proteínas con localización extracelular, péptidos señal, y epítopenos de células B. A continuación, estos genes se filtran de los atributos deseables que harían buena vacuna objetivos tales como exteriores proteínas de la membrana. Esas proteínas y luego se someten a pruebas normales de laboratorio húmedo para la respuesta inmune.

Aunque el uso de la tecnología bioinformática para desarrollar vacunas se ha convertido en típico en los últimos diez años, los laboratorios generales a menudo no tienen el software de avanzada que puede hacer esto. Sin embargo, hay un número creciente de programas que hacen la información vacunología inversa más accesible.

Proyectos de bioinformática asistido por ordenador se están convirtiendo en muy popular, ya que ayudan a guiar a los experimentos de laboratorio.

1.7.5 Arquitectura cliente - servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La red cliente-servidor es una red de comunicaciones en la cual los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones

con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

1.7.6 HTML

HTML, siglas de *HyperText Markup Language* («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Es el lenguaje con el que se definen las páginas web.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, etc.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.). Sin embargo, para interpretar

correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Usualmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Un navegador no actualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple.

1.7.7 PHP

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

CURL

cURL es una herramienta para usar en un intérprete de comandos para transferir archivos con sintaxis URL, soporta FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, FILE y LDAP. cURL soporta certificados HTTPS, HTTP POST, HTTP PUT, subidas FTP, Kerberos, subidas mediante formulario HTTP, proxies, cookies, autenticación mediante usuario+contraseña (Basic, Digest, NTLM y Negotiate para HTTP y kerberos4 para FTP), continuación de transferencia de archivos, tunneling de proxy http y muchas otras prestaciones. cURL es open source/software libre distribuido bajo la Licencia MIT.

El principal propósito y uso para cURL es automatizar transferencias de archivos o secuencias de operaciones no supervisadas. Es por ejemplo una buena herramienta para simular las acciones de un usuario en un navegador web. Libcurl es la biblioteca/API correspondiente que los usuarios pueden incorporar en sus programas; cURL actúa como un envoltorio (wrapper) aislado para la biblioteca libcurl. libcurl se usa para proveer capacidades de transferencia de URL a numerosas aplicaciones, tanto libres y open source como así también privativas. La biblioteca "libcurl" se puede usar desde más de 30 lenguajes distintos.

1.7.8 Javascript

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y

páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

1.7.9 Expresiones regulares

Una expresión regular, a menudo llamada también regex, es una secuencia de caracteres que forma un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones.

En el área de la programación las expresiones regulares son un método por medio del cual se pueden realizar búsquedas dentro de cadenas de caracteres. Sin importar si la búsqueda requerida es de dos caracteres en una cadena de 10 o si es necesario encontrar todas las apariciones de un patrón definido de caracteres en un archivo de millones de caracteres, las expresiones regulares proporcionan una solución para el problema. Adicionalmente, un uso derivado de la búsqueda de patrones es la validación de un formato específico en una cadena de caracteres dada, como por ejemplo fechas o identificadores.

1.7.10 NetMHCpan

Servidor NetMHCpan predice unión de los péptidos a cualquier molécula MHC conocida usando redes neuronales artificiales (RNAs). El método está entrenado en más de 150.000 datos cuantitativos vinculantes que abarcan más de 150 moléculas de MHC diferentes. Las predicciones se pueden hacer para HLA-A, B, C, E y G alelos, así como para los primates no-humanos, ratón, ganado vacuno y porcino. Además, el usuario puede cargar longitud del MHC secuencias de proteínas completas, y que el servidor predecir restringidas al MHC de péptidos a partir de cualquier proteína de interés dada.

1.7.11 NetMHCIIPan

Servidor NetMHCIIPan predice unión de péptidos a moléculas MHC de clase II. Las predicciones están disponibles para todos los tres isotipos humanos MHC de clase II, HLA-DR, HLA-DP y HLA-DQ, así como moléculas de ratón. NetMHCIIPan puede producir predicciones para los péptidos de 9 - 19 aminoácidos de longitud.

Dos tipos de presentación se manejan en lista de los péptidos o una secuencia de proteína en formato FASTA. El servidor proporciona una posibilidad para que el usuario elija molécula de MHC en cuestión a partir de una larga lista de alelos o alternativamente cargar la secuencia de la proteína del MHC de interés. Para HLA-DP y HLA-DQ cadenas alfa y beta moléculas se pueden elegir por separado de la lista. También, para HLA-DP, HLA-DQ y moléculas de ratón, secuencias de longitud completa de las cadenas alfa y beta deben ser cargados por separado. Para HLA-DR sólo secuencia de la cadena beta se debe cargar.

1.7.12 Cello

CELLO es un predictor de localización subcelular utilizando una clasificación de SVM multi-clase. Utiliza 4 tipos de esquemas de secuencia de codificación: la composición de aminoácidos, la composición di-péptido, la composición de aminoácidos repartió y la secuencia de composición en base a las propiedades físico-químicas de aminoácidos.

Combina los votos de estos clasificadores y utiliza los votos del jurado para determinar la asignación final.

1.7.13 BepiPred

Servidor BepiPred predice la localización de epítopos lineales de células B utilizando una combinación de un modelo oculto de Markov y un método de escala de propensión.

1.7.14 Formato FASTA

En bioinformática, el formato FASTA es un formato de fichero informático basado en texto, utilizado para representar secuencias bien de ácidos nucleicos, bien de péptido, y en el que los pares de bases o los aminoácidos se representan usando códigos de una única letra. El formato también permite incluir nombres de secuencias y comentarios que preceden a las secuencias en sí.

La simplicidad del formato FASTA hace fácil el manipular y analizar secuencias usando herramientas de procesado de textos y lenguajes de guion como Python y PERL.

Una secuencia bajo formato FASTA comienza con una descripción en una única línea (línea de cabecera), seguida por líneas de datos de secuencia. La línea de descripción se distingue de los datos de secuencia por un símbolo '>' (mayor que) en la primera columna. La palabra siguiente a este símbolo es el identificador de la secuencia, y el resto de la línea es la descripción (ambos son opcionales). No debería existir espacio entre el '>' y la primera letra del identificador. Se recomienda que todas las líneas de texto sean menores de 80 caracteres. La secuencia termina si aparece otra línea comenzando con el símbolo '>'; esto indica el comienzo de otra secuencia. Un ejemplo simple de una secuencia en el formato FASTA puede ser:

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAI PYIGTNLVEW
IWGGFSVDKATLNRRFAFHILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFGLLLIL
LLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVILGLMPFLHTS
KHRSMMLRPLSQALFWTLTMDLLTWIGSQPV EYPYTIIGQM ASILYFSIIAFLPIAGXIENY
```

1.8 Procedimiento y descripción de las actividades realizadas

1.8.1 Selección de las herramientas inmunoinformáticas a integrar

El primer paso fue la selección de las herramientas inmunoinformáticas. Cada una de propósito diferente, pero que en conjunto son para un mismo fin (determinar las proteínas candidatas para vacunas).

Platicando con los investigadores interesados se decidió seleccionar cuatro herramientas, por ser una cantidad viable considerando el tiempo disponible para el proyecto y por la relevancia de sus resultados en conjunto.

Las herramientas seleccionadas fueron:

- Cello v.2.5 ([Anexo 1](#))
- NetMHCpan v2.8 ([Anexo 2](#))
- NetMHCIIPan v3.0 ([Anexo 3](#))
- BepiPred v1.0 ([Anexo 4](#))

Las cuales cubren respectivamente las necesidades de:

- Localización subcelular
- Unión de péptidos a moléculas MHC I
- Unión de péptidos a moléculas MHC II
- Localización de epítopos de células B

1.8.2 Integración de interfaces

Lo siguiente fue integrar en una sola aplicación las herramientas seleccionadas. Primeramente fue necesario analizar el código HTML y JavaScript de cada herramienta, con la intención de poder llevar a cabo la integración de la forma más limpia posible.

El primer paso consistió en unir los cuatro formularios en uno solo, de esta forma el usuario no tendrá que dar cuatro clics de enviar. Solo dará un clic, y REVXINE de forma interna redireccionará a cada servidor el POST que le corresponde.

Al unir los campos en un solo formulario, se identificó que algunos tienen el mismo nombre y pertenecen a servidores diferentes, lo cual ocasiona conflictos. Por lo tanto, se decidió dar nombres únicos a cada campo, de forma que sea posible identificar a qué servidor pertenecen.

Así también, se dejó solo un campo en caso de encontrar repetidos con el mismo propósito, tales como:

- SEQPASTE, es un cuadro de texto para capturar las secuencias en formato FASTA.
- SEQSUB, es un tipo file, para para subir un archivo con las secuencias en formato FASTA.
- inp, un select para seleccionar el formato de las secuencias (FASTA por default y como única opción para esta versión de REVXINE).

En NetMHCpan y NetMHCIIPan existe un campo llamado “sort” que sirve para activar el despliegue ordenado de los resultados. Este campo fue eliminado de la interfaz REVXINE, pero se considera en el armado de POST interno y se le da por default “on”. Se decidió así, porque el ordenamiento de los resultados es indispensable para el parseo que se explicará más adelante.

En la siguiente tabla se muestran todos los campos del formulario HTML. El nombre original con el que se envía a cada servidor, el nombre temporal para que puedan ser parte del mismo formulario, el tipo del campo, y el servidor al cual irá dirigido el POST generado de forma interna por REVXINE.

	name original	name temporal	Tipo	Servidor
1	fasta, SEQPASTE, SEQPASTE, SEQPASTE	SEQPASTE	textarea	Cello, NetMHCpan, NetMHCIIPan, BepiPred
2	file, SEQSUB, SEQSUB, SEQSUB	SEQSUB	input file	Cello, NetMHCpan, NetMHCIIPan, BepiPred
3	inp	inp	select	NetMHCpan,

				NetMHCIIpan
4	seqtype	Cello_seqtype	input radio	Cello
5	species	Cello_species	input radio	Cello
6	Submit	Submit	input submit	Cello
7	configfile	NetMHCII_configfile	input hidden	NetMHCpan
8	len	NetMHCII_len	select	NetMHCpan
9	master	NetMHCII_master	select	NetMHCpan
10	slave0	NetMHCII_slave0	select	NetMHCpan
11	allele	NetMHCII_allele	input text	NetMHCpan
12	MHCSEQPASTE	NetMHCII_MHCSEQPASTE	textarea	NetMHCpan
13	MHCSEQSUB	NetMHCII_MHCSEQSUB	input file	NetMHCpan
14	thrs	NetMHCII_thrs	input text	NetMHCpan
15	thas	NetMHCII_thas	input text	NetMHCpan
16	thrw	NetMHCII_thrw	input text	NetMHCpan
17	thaw	NetMHCII_thaw	input text	NetMHCpan
18	IC50	NetMHCII_IC50	input checkbox	NetMHCpan
19	configfile	NetMHCII_configfile	input hidden	NetMHCIIpan
20	length	NetMHCII_length	input text	NetMHCIIpan
21	master	NetMHCII_master	select	NetMHCIIpan
22	slaveA	NetMHCII_slaveA	select	NetMHCIIpan
23	slaveB	NetMHCII_slaveB	select	NetMHCIIpan
24	allele	NetMHCII_allele	input text	NetMHCIIpan
25	MHCSEQPASTEb	NetMHCII_MHCSEQPASTEb	textarea	NetMHCIIpan
26	MHCSEQSUBa	NetMHCII_MHCSEQSUBa	input file	NetMHCIIpan
27	MHCSEQPASTEb	NetMHCII_MHCSEQPASTEb	textarea	NetMHCIIpan
28	MHCSEQSUBb	NetMHCII_MHCSEQSUBb	input file	NetMHCIIpan
29	thrs	NetMHCII_thrs	input text	NetMHCIIpan
30	thas	NetMHCII_thas	input text	NetMHCIIpan
31	thrw	NetMHCII_thrw	input text	NetMHCIIpan
32	thaw	NetMHCII_thaw	input text	NetMHCIIpan
33	filt	NetMHCII_filt	select	NetMHCIIpan
34	thrf	NetMHCII_thrf	input text	NetMHCIIpan
35	thaf	NetMHCII_thaf	input text	NetMHCIIpan
36	fast	NetMHCII_fast	input checkbox	NetMHCIIpan
37	unique	NetMHCII_unique	input checkbox	NetMHCIIpan
38	threshold	BepiPred_threshold	input text	BepiPred
39	configfile	BepiPred_configfile	input hidden	BepiPred

Además en el mismo formulario existen más campos definidos, que sirven para la gestión de los procesos dentro de REVXINE

	name	Tipo	Descripción
40	LOCATION[]	select	Opción de filtrado para cello
41	RELIABILITY	radio	Nivel de filtrado para cello
42	Cello	checkbox	Activar el uso de Cello
43	NetMHCI	checkbox	Activar el uso de NetMHCI
44	NetMHCII	checkbox	Activar el uso de NetMHCII
45	BepiPred	checkbox	Activar el uso de BepiPred

Una vez modificado los campos se procedió a unir todas las secciones y dar diseño a la interfaz de REVXINE. La interfaz integradora de herramientas se muestra en la imagen de abajo.

The screenshot shows the REVXINE v1.0 web application interface. At the top, there is a header with the logo of the Centro de Investigaciones Biológicas del Noroeste, S.C. and the text "REVXINE v1.0 : Conjunto de herramientas para vacunología inversa". Below the header, there is a navigation menu with links to "Inicio", "Instrucciones", "Servidores", and "Contacto".

The main area contains several input fields and dropdown menus:

- Tipos de entrada:** A dropdown menu set to "Fasta" (1).
- Pegue una sola secuencia o varias secuencias en formato FASTA en el campo de abajo:** A text area containing a FASTA sequence (2).
- Seleccionar archivo:** A button indicating no file was selected (3).
- Localización Subcelular:** A dropdown menu with "Cello" checked (4). It includes options like "Endocellular" (5), "Membrane", "CellWall", and "OuterMembrane". A radio button for "Confabilidad" is set to "Normal" (6).
- Unión molecular MHCI:** A checkbox checked (7).
- Unión molecular MHCII:** A checkbox checked (8).
- Localización de epítopos lineales de células B:** A checkbox checked (9).
- Submit:** A button at the bottom left (10).

At the bottom of the page, there is footer text: "Para mas información visite la sección [SERVIDORES](#)" and the copyright information "Instituto Tecnológico de La Paz" and "Centro de Investigaciones Biológicas del Noroeste, S.C."

La sección del lado izquierdo es la interfaz principal de REVXINE, los campos están numerados y se describen a continuación:

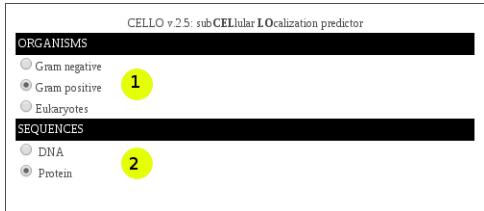
1. Elegir el formato de las secuencias de aminoácidos.
2. Escribir las secuencias de aminoácidos en formato FASTA
3. Subir un archivo en texto plano, con las secuencias de aminoácidos en formato FASTA.
4. Habilita el uso de Cello v.2.5, para realizar filtrado de las secuencias de aminoácidos.
5. Selecciona la localización subcelular de interés, para realizar experimentos sólo con las secuencias de aminoácidos en esa localización.
6. Nivel de holgura en el filtrado.
7. Habilita uso de NetMHCpan v2.8, para buscar péptidos de unión a MHCI.
8. Habilita uso de NetMHCIIPan v3.0, para buscar péptidos de unión a MHCI.
9. Habilita uso de BepiPred v1.0, para buscar epítópos de unión a células B.
10. Envía la solicitud a REVXINE para el procesamiento y análisis de las secuencias de aminoácidos.

Los campos 4,7,8,9, tienen un hipervínculo que despliega los parámetros particulares de cada servidor, en el recuadro de la derecha.

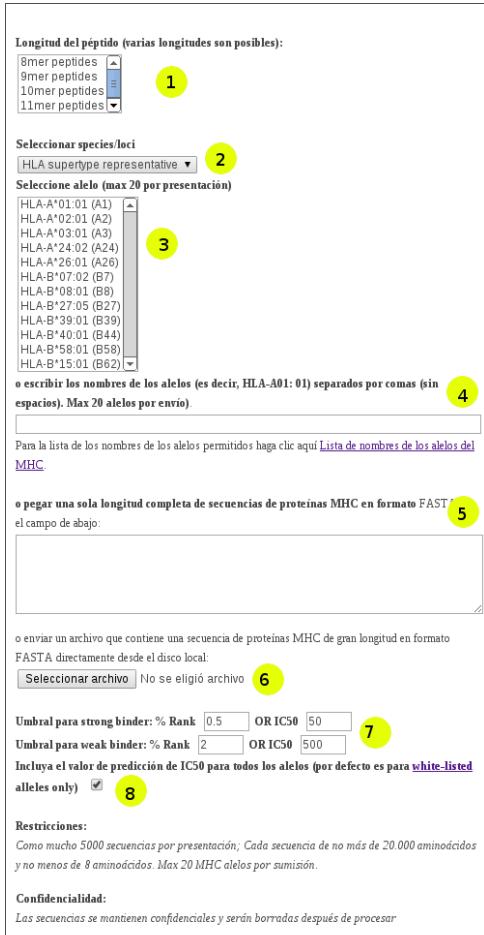
The screenshot shows the REVXINE v1.0 web application. On the left, there's a sidebar with the logo of the Centro de Investigaciones Biológicas del Noroeste, S.C., and links for Home, Instructions, Servers, and Contact. The main content area has a title "RENXINE v1.0 : Conjunto de herramientas para vacunología inversa". Below it, a form titled "PRESENTAR SECUENCIA(S)" has a dropdown menu set to "FastA". A text input field contains a FASTA sequence: "IAMPTRKPPSQTLTRIEAPCPCHVAKTERAVVTPAPEVADQLPPPTPAVPSKFGGLWILAVVASHGSSPAISPKCSPSAUCQSSSTAPCPLPSPSPSWSPACDSLAPLTSHSAPGYPAVHDPSSYSAVCGTHIKALPVPMKSSKFVLPPSVFAISCPPLPHPCDLSLTTPS". Below the input field is a note: "a enviar un archivo en formato FASTA directamente desde su disco local". There's also a link "Seleccionar archivo" and a note "(No se envió archivo)". Under "ANALIZAR SECUENCIA(S) CON:", there's a section titled "Haz clic en el nombre para modificar los parámetros:" with several checkboxes and dropdown menus. One dropdown menu under "Localización Subcelular" is expanded, showing options: Mitochondria, Membrane, Cellwall, OuterMembrane, with "Mitochondria" selected. Other sections include "Unión molecular MHCI" (checkboxes for "NetMHCI" and "NetMHCIIPan"), "Unión molecular MHCI" (checkbox for "BepiPred"), and "Localización de epitope/línea de célula B" (checkbox for "BepiPred"). At the bottom, there's a "Submit" button and a note "Para más información visite la sección [PEPTIDOS/EP](#)". To the right of the main form, a large yellow rectangular area is labeled "Sección de parámetros particulares".

El contenido de la sección de formatos particulares se muestra a continuación ordenado por servidor:

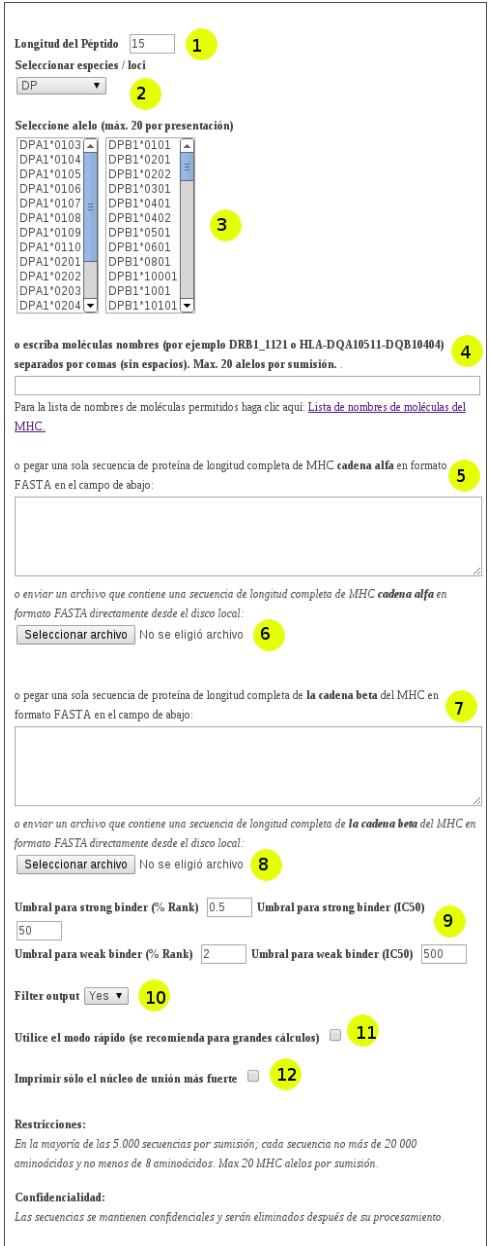
Cello

Contenido	Descripción
	<ol style="list-style-type: none"> 1. Tipo de organismo de la secuencia 2. Tipo de secuencia

NetMHCpan

Contenido	Descripción
	<ol style="list-style-type: none"> 1. Longitud del péptido. 2. Selección de familia de alelo. 3. Selección de alelo. 4. Alelos Seleccionados. 5. Secuencia de proteínas MHC en formato FASTA 6. Secuencia de proteínas MHC en formato FASTA desde archivo 7. Opciones de umbral para consideración de unión fuerte y débil. 8. Opción de activar IC50 para todos los alelos

NetMHCIIpan

Contenido	Descripción
 <p>The screenshot shows the user interface for NetMHCIIpan. It includes fields for peptide length (15), species/locus selection (DP), a dropdown for allele selection (DPA1*0103 selected), a list of alleles (DPA1*0103-DPB1*10101), a text area for molecule names (empty), a FASTA sequence input (empty), an archive selection button (No se eligió archivo), a beta chain sequence input (empty), a FASTA archive selection button (No se eligió archivo), filter settings (strong and weak binder thresholds), a rapid mode checkbox (unchecked), and a print option checkbox (unchecked). There are also sections for restrictions and confidentiality.</p> <ol style="list-style-type: none"> 1. Longitud del Péptido: 15 2. Seleccionar especies / loci: DP 3. Selección alelo (máx. 20 por presentación): DPA1*0103 4. o escriba moléculas nombres (por ejemplo DRB1_1121 o HLA-DQA10511-DQB10404) separados por comas (sin espacios). Max. 20 alelos por sumisión. 5. o pegar una sola secuencia de proteína de longitud completa de MHC cadena alfa en formato FASTA en el campo de abajo: 6. Seleccionar archivo: No se eligió archivo 7. o pegar una sola secuencia de proteína de longitud completa de la cadena beta del MHC en formato FASTA en el campo de abajo: 8. Seleccionar archivo: No se eligió archivo 9. Umbral para strong binder (% Rank): 0.5 Umbral para strong binder (IC50): 50 10. Umbral para weak binder (% Rank): 2 Umbral para weak binder (IC50): 500 11. Filter output: Yes 12. Utilice el modo rápido (se recomienda para grandes cálculos): <input type="checkbox"/> 13. Imprimir sólo el núcleo de unión más fuerte: <input type="checkbox"/> 14. Restricciones: En la mayoría de las 5.000 secuencias por sumisión, cada secuencia no más de 20 000 aminoácidos y no menos de 8 aminoácidos. Max 20 MHC alelos por sumisión. 15. Confidencialidad: Las secuencias se mantienen confidenciales y serán eliminados después de su procesamiento. 	<ol style="list-style-type: none"> 1. Longitud de péptido. 2. Selección de familia de alelo. 3. Selección de alelo. 4. Alelos seleccionados. 5. Secuencia de proteína MHC cadena alfa. 6. Secuencia de proteína MHC cadena alfa desde archivo. 7. Secuencia de proteína MHC cadena beta. 8. Secuencia de proteína MHC cadena beta desde archivo. 9. Opciones de umbral para consideración de unión fuerte y débil. 10. Opción de filtrado. 11. Opción de modo rápido. 12. Opción de imprimir solo núcleo de unión más fuerte.

Contenido	Descripción
Puntuación de umbral para la asignación epítopo 0.35 1	1. Umbral para la asignación de epítopos

1.8.3 Comunicación con los servidores

Los servidores inmunoinformáticos seleccionados, solo aceptan POSTs en su formato original. Es por ello que el submit de nuestro formulario llama a la página “result.php” la cual se encarga de descomponer el POST en cuatro POSTs fieles a su respectivo servidor.

Antes de realizar la descomposición, se desarrolló el siguiente método que fue esencial para poder generar los POST individuales:

```
string multipart_build_query(array $fields, string $boundary);
```

El método multipart_build_query genera una cadena de contenido multipart, a partir de un arreglo en PHP. Esta misma cadena es la que se utiliza para enviar como POST al servidor por medio de HTTP utilizando la librería cURL, eso lo veremos más adelante.

```
function multipart_build_query($fields, $boundary){
    $retval = '';
    foreach($fields as $key => $value){
        $retval .= "--$boundary\r\nContent-Disposition: form-data;
name=\"$key\"\r\n\r\n$value\r\n";
    }
    $retval .= "--$boundary--";
    return $retval;
}
```

Por ahora los cuatro servidores utilizan el formato multiparte para sus POSTs.

Una vez programado el método para generar los POSTs en formato multipart. Se programó la descomposición del POST temporal para formar los POSTs respectivos para cada servidor, renombrando cada campo con su nombre original.

Para realizar la descomposición se utilizó el método:

```
void generatePost(string $fasta, array $e_post, array $files, string $boundary);
```

Cada servidor tiene su clase con atributos y métodos tal y como se muestra en el diagrama de clases ([Anexo 6](#)). El método generatePost está declarado en todas las clases servidor por igual, pero difieren en su contenido. A continuación se muestra el método generatePost de cada clase servidor.

<i>Cello</i>	<pre>public function generatePost(\$fasta,\$e_post,\$files,\$boundary){ if(isset(\$e_post['Cello'])){ \$array_Cello=array('species'=>\$e_post['Cello_species'], 'seqtype' => \$e_post['Cello_seqtype'], 'fasta'=>\$fasta, 'Submit' => 'Submit'); \$this->setPost(multipart_build_query(\$array_Cello, \$boundary)); } }</pre>
<i>NetMHCpan</i>	<pre>public function generatePost(\$fasta,\$e_post,\$files,\$boundary){ \$array_NetMHC; if(isset(\$e_post['NetMHC'])) { \$array_NetMHC=array('inp' => \$e_post['inp'], 'SEQPASTE'=>\$fasta, 'configfile' => \$e_post['NetMHC_configfile'], 'len'=>(isset(\$e_post['NetMHC_len']))?\$e_post['NetMHC_len']: '', 'master' => \$e_post['NetMHC_master'], 'slave0' => (isset(\$e_post['NetMHC_slave0']))?\$e_post['NetMHC_slave0']: '', 'allele' => \$e_post['NetMHC_allele'], 'MHCSEQPASTE' => \$e_post['NetMHC_MHCSEQPASTE'], 'thrs' => \$e_post['NetMHC_thrs'], 'thas' => \$e_post['NetMHC_thas'], 'thrw' => \$e_post['NetMHC_thrw'], 'thaw' => \$e_post['NetMHC_thaw'], 'IC50' => (isset(\$e_post['NetMHC_IC50']))?\$e_post['NetMHC_IC50']: '', 'sort' => 'on'); } if(isset(\$files['NetMHC_MHCSEQSUB']))&&empty(\$files['NetMHC_MHCSEQSUB']['tmp_name'])) }</pre>

```

$array_NetMHCII['MHCSEQPASTE']=file_get_contents($files['NetMHCII_MHCSEQSUB'][ 'tmp_name'], true);

$this->setPost(multipart_build_query($array_NetMHCII, $boundary));
}

NetMHCIIpan
public function generatePost($fasta,$e_post,$files,$boundary){
$array_NetMHCII;
if(isset($e_post['NetMHCII'])) {
$array_NetMHCII=array(
'inp' => $e_post['inp'],
'SEQPASTE'=>$fasta,
'configfile' => $e_post['NetMHCII_configfile'],
'length' => $e_post['NetMHCII_length'],
'master' => $e_post['NetMHCII_master'],
'slaveA' => (isset($e_post['NetMHCII_slaveA']))?$e_post['NetMHCII_slaveA']: '',
'slaveB' => (isset($e_post['NetMHCII_slaveB']))?$e_post['NetMHCII_slaveB']: '',
'allele' => $e_post['NetMHCII_allele'],
'MHCSEQPASTEb' => $e_post['NetMHCII_MHCSEQPASTEb'],
'MHCSEQPASTEb' => $e_post['NetMHCII_MHCSEQPASTEb'],
'thrs'=> $e_post['NetMHCII_thrs'],
'thas'=> $e_post['NetMHCII_thas'],
'thrw'=> $e_post['NetMHCII_thrw'],
'thaw'=> $e_post['NetMHCII_thaw'],
'filt'=> $e_post['NetMHCII_filt'],
'thrf'=> $e_post['NetMHCII_thrf'],
'thaf'=> $e_post['NetMHCII_thaf'],
'fast'=> (isset($e_post['NetMHCII_fast']))?$e_post['NetMHCII_fast']: '',
'unique'=> (isset($e_post['NetMHCII_unique']))?$e_post['NetMHCII_unique']: '',
'sort' => 'on'
);

if(isset($files['NetMHCII_MHCSEQSUBa'])&&empty($files['NetMHCII_MHCSEQSUBa'][ 'tmp_name'])) {
$array_NetMHCII['MHCSEQPASTEb']=file_get_contents($files['NetMHCII_MHCSEQSUBa'][ 'tmp_name'], true);
if(isset($files['NetMHCII_MHCSEQSUBb'])&&empty($files['NetMHCII_MHCSEQSUBb'][ 'tmp_name'])) {
$array_NetMHCII['MHCSEQPASTEb']=file_get_contents($files['NetMHCII_MHCSEQSUBb'][ 'tmp_name'], true);
}

$this->setPost(multipart_build_query($array_NetMHCII, $boundary));
}
}

BepiPred
public function generatePost($fasta,$e_post,$files,$boundary){
if(isset($e_post['BepiPred'])) {
$array_BepiPred=array(
'SEQPASTE'=>$fasta,
'configfile' => $e_post['BepiPred_configfile'],
'threshold'=>$e_post['BepiPred_threshold']
);
$this->setPost(multipart_build_query($array_BepiPred, $boundary));
}
}

```

Ya que esta generado el post con los nombres correctos y en formato multiparte. Es posible enviarlo desde PHP al servidor correspondiente, utilizando una librería para mandar y recibir paquetes HTTP desde PHP llamada cURL. Esta librería es muy completa y permite realizar muchas operaciones de las cuales no se limita únicamente a HTTP.

En nuestra herramienta, utilizamos cURL para enviar un paquete POST como solicitud desde PHP a cada servidor, y solicitar el HTML de cada servidor hasta que el servidor haya desplegado un resultado.

Para realizar todo lo anterior se programaron los siguientes métodos:

```
array curl_request(string $url);
array curl_sendPost(string $url, array $post, string $boundary);
array waitResult(array $http, string $status_patron, int $sleep_min, int $sleep_max);
```

- El método curl_request solicita el html de una url.
- El método curl_sendPost se encarga de enviar el post generado por generatePost.
- El método waitResult utiliza curl_request de manera indefinida y se detiene cuando el patrón de status en el html desaparezca.

A continuación se muestra el código php de los tres métodos:

```
function curl_request($url) {
    $ch = curl_init ( $url );
    $options = array (
        CURLOPT_HEADER => false,
        CURLOPT_USERAGENT => "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.1)
Gecko/20061204 Firefox/2.0.0.1",
        CURLOPT_URL => $url,
        CURLOPT_RETURNTRANSFER => true
    );
    curl_setopt_array ( $ch, $options );
    $html = curl_exec ( $ch );

    $httpResult = curl_getinfo ( $ch );

    curl_error ( $ch );
    curl_close ( $ch );
```

```

$http = array (
    'url' => $httpResult ['url'],
    'html' => $html
);

return $http;
}

function waitResult($http, $status_patron, $sleep_min, $sleep_max) {
do {
    $url = $http ['url'];
    unset ( $http );
    sleep ( mt_rand ( $sleep_min, $sleep_max ) );
    $http = curl_request ( $url );
} while ( preg_match ( $status_patron, $http ['html'] ) );

return $http;
}

function curl_sendPost($url, $post, $boundary) {
$ch = curl_init ( $url );
$options = array (
    CURLOPT_HEADER => false,
    CURLOPT_USERAGENT => "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.1)
Gecko/20061204 Firefox/2.0.0.1",
    CURLOPT_URL => $url,
    CURLOPT_HTTPHEADER => array (
        "Content-Type: multipart/form-data; boundary=$boundary"
    ),
    CURLOPT_POST => true,
    CURLOPT_POSTFIELDS => $post,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_RETURNTRANSFER => true
);

curl_setopt_array ( $ch, $options );
$html = curl_exec ( $ch );

$httpResult = curl_getinfo ( $ch );

curl_error ( $ch );
curl_close ( $ch );

$http = array (
    'url' => $httpResult ['url'],
    'html' => $html
);

return $http;
}

```

Cuando estos métodos terminan su función REVXINE ya tiene en sus manos los resultados de cada servidor de acuerdo a la solicitud del usuario. El siguiente paso es validar que los resultados sean correctos. Dicho de otra forma, el siguiente paso es revisar que el servidor efectivamente logró el análisis de las secuencias que el usuario solicitó.

Para realizar la validación fue necesario programar el siguiente método en cada clase de servidores:

```
bool resultIsValid(string $html);
```

- El método resultIsValid se encarga de validar el documento html que presenta cada servidor como resultado de su análisis. La validación consiste en buscar el patrón de resultado válido dentro de cada documento html.

Código:

```
public function resultIsValid($html,$patronIsOk) {  
    if (preg_match ( $this->patron , $html ))  
        return TRUE;  
    else  
        return FALSE;  
}
```

El atributo patrón para cada clase servidor es el siguiente:

```
Cello  
public $patron = "/CELLO [pP]rediction:/";  
NetMHCpan  
public $patron = "/pos[ ]+HLA[ ]+peptide[ ]+Identity[ ]+1-log50k\(\aff\)[ ]+Affinity\(\nM\)[ ]+%Rank[ ]+BindLevel/";  
NetMHCIIpan  
public $patron = "/pos[ ]+Allele[ ]+peptide[ ]+Identity[ ]+Pos[ ]+Core[ ]+1-log50k\(\aff\)[ ]+Affinity\(\nM\)[ ]+%Rank[ ]+BindingLevel/";  
BepiPred  
public $patron = "/seqname[ ]+source[ ]+feature[ ]+start[ ]+end[ ]+score[ ]+N\|A[ ]+\?/";
```

La integración de los servidores termina al desplegar los resultados de cada uno. En caso de que existiese un error con el servidor, REVXINE despliega el error encontrado al procesar la solicitud con el servidor involucrado. Los errores son gestionados por el siguiente arreglo que es declarado global para todas las clases servidores y determina el estado del resultado de la petición:

```
$ERRORS = array (
    0 => 'is ok',
    1 => 'Error: El servidor no pudo resolver la peticion',
    2 => 'Error: El servidor fue interrumpido',
    3 => 'Error: Imposible acceder al servidor',
    4 => 'Error: El filtrado de CELLO fue excesivo.'
);
```

Después de haber validado los resultados, REVXINE muestra la interfaz de resultados donde se visualizan dos secciones, la primera de lado izquierdo es el menú de los servidores, y en la derecha se muestran los resultados de dicho servidor.

CELLO RESULTS		
SegID: MAP0858	Analysis Report	
SVM	LOCALIZATION	RELIABILITY
Amino Acid Comp.	Extracellular	0.377
N-peptide Comp.	Extracellular	0.544
Partitioned seq Comp.	Membrane	0.445
Physico-Chemical Comp.	Membrane	0.827
Neighboring seq Comp.	Membrane	0.737
CELLO Predictions:	Membrane	2.050 *
	Extracellular	1.634
	Cytoplasm	0.638
	CellWall	0.078

[\(Home \)](#) [\(Documentation \)](#)

Instituto Tecnológico de La Paz
Centro de Investigaciones Biológicas del Noroeste, S.C.

Resultados

NetMHCI

Error: Imposible acceder al servidor

Instituto Tecnológico de La Paz
Centro de Investigaciones Biológicas del Noroeste, S.C.

Hasta este punto termina la integración de las herramientas, siendo REVXINE un intermediario entre el usuario y los servidores inmunoinformáticos.

1.8.4 Parseo de respuestas

El parseo de las respuestas consiste en leer el html resultado, y construir una estructura de datos que permita acceder a cada dato relevante considerado en el html. Para ello, se siguió la estrategia utilizada en la validación del resultado, la cual consiste en buscar patrones que nos permitan ubicar los datos de interés.

Se programó una clase parse para cada servidor, con el fin de que en ella esté todo el procesamiento en cuanto al parse y generación de los reportes para su respectivo servidor.

El constructor recibe el resultado html del servidor y genera una estructura con arreglos multidimensionales. Esta estructura contiene todos los elementos del html resultado, pero ahora son accesibles fácilmente.

El código de las clases parse de cada servidor se encuentra en ([Anexo 7](#)). A continuación se muestran las hojas de resultado de cada servidor, y el arreglo multidimensional que genera el parse de su respectivo servidor. Para estos ejemplos se utilizaron las secuencias “seq1” y “MAP1025”. Seq1 para NetMHCI, NetMHCII , BepiPred, y MAP1025 para Cello.

Secuencias de prueba en formato FASTA:

```
>seq1
ASTPGHTIIFYAVCLHNDRTTIP
```

```
>MAP1025
MPMTDQPPPGGAYPPPPSSPGSPGGQPTPHPGGQQPPPPPGGSYPPPPPGGSY
PPPPPSGGYAPPPPGLPAIRTLPTQDYTPWLTRALAFVIDILPYVVVHGIGTAILVATQ
QTACITDVTQYAVNQYCATIONSTLGLVAQWLASIVGLFYLIWNYGYRQGTTGSSVGK
SVMKFKVSEVTGQPVGFGMSVRAHALFVDIAICFIGFLPLWDSKRQTLADKIMTT
VCLPLDGSESPPS
```

Hoja de resultados de Cello con secuencia MAP1025.

CELLO RESULTS

SeqID: MAP1025

Analysis Report:

SVM	LOCALIZATION	RELIABILITY
Amino Acid Comp.	Periplasmic	0.287
N-peptide Comp.	InnerMembrane	0.543
Partitioned seq. Comp.	Cytoplasmic	0.357
Physico-chemical Comp.	Extracellular	0.390
Neighboring seq. Comp.	Periplasmic	0.335

CELLO Prediction:

	LOCALIZATION	RELIABILITY
InnerMembrane	1.474 *	
Periplasmic	1.308 *	
Extracellular	1.233 *	
Cytoplasmic	0.696	
OuterMembrane	0.289	

[Home | Documentation]

Estructura de datos, generada por la clase Cello_parse.

```
Array
(
    [0] => result
    [1] => seqid
    [2] => report
    [3] => prediction
    [4] => bests
    [5] => best
    [result] => SeqID: MAP1025Analysis Report:SVMLOCALIZATIONRELIABILITYAmino Acid
Comp.Periplasmic0.287N-peptide Comp.InnerMembrane0.543Partitioned seq.
Comp.Cytoplasmic0.357Physico-chemical Comp.Extracellular0.390Neighboring seq.
Comp.Periplasmic0.335CELLO
Prediction:InnerMembrane1.474*Periplasmic1.308*Extracellular1.233*Cytoplasmic0.696OuterMembran
e0.289*****
    [seqid] => MAP1025
    [report] => Array
        (
            [Amino Acid Comp.] => Array
                (
                    [localization] => Periplasmic
                    [reliability] => 0.287
                )
            [N-peptide Comp.] => Array
                (
                    [localization] => InnerMembrane
                    [reliability] => 0.543
                )
            [Partitioned seq. Comp.] => Array
                (
                    [localization] => Cytoplasmic
                    [reliability] => 0.357
                )
            [Physico-chemical Comp.] => Array
                (
                    [localization] => Extracellular
                    [reliability] => 0.390
                )
        )
)
```

```

        )
    [Neighboring seq. Comp.] => Array
    (
        [
            [localization] => Periplasmic
            [reliability] => 0.335
        ]
    )
[prediction] => Array
(
    [
        [0] => Array
        (
            [
                [localization] => InnerMembrane
                [reliability] => 1.474*
            ]
        )
        [1] => Array
        (
            [
                [localization] => Periplasmic
                [reliability] => 1.308*
            ]
        )
        [2] => Array
        (
            [
                [localization] => Extracellular
                [reliability] => 1.233*
            ]
        )
        [3] => Array
        (
            [
                [localization] => Cytoplasmic
                [reliability] => 0.696
            ]
        )
        [4] => Array
        (
            [
                [localization] => OuterMembrane
                [reliability] => 0.289
            ]
        )
    ]
)
[bests] => Array
(
    [
        [0] => Array
        (
            [
                [localization] => InnerMembrane
                [reliability] => 1.474
            ]
        )
        [1] => Array
        (
            [
                [localization] => Periplasmic
                [reliability] => 1.308
            ]
        )
        [2] => Array
        (
            [
                [localization] => Extracellular
                [reliability] => 1.233
            ]
        )
    ]
)
[best] => Array
(
    [
        [localization] => InnerMembrane
        [reliability] => 1.474
    ]
)
)

```

Hoja de resultados de NetMHCpan con proteína Seq1 y alelo HLA-A01:01.

NetMHCpan Server - prediction results

Technical University of Denmark

```
# NetMHCpan version 2.8
# Input is in FSA format
# Peptide length 9

HLA-A01:01 : Estimated prediction accuracy 0.853 (using nearest neighbor HLA-A01:01)

# Affinity Threshold for Strong binding peptides 50.000
# Affinity Threshold for Weak binding peptides 500.000
# Rank Threshold for Strong binding peptides 0.500
# Rank Threshold for Weak binding peptides 2.000

pos HLA peptide Identity 1-log50k(aff) Affinity(nM) %Rank BindLevel
-----
```

pos	HLA	peptide	Identity	1-log50k(aff)	Affinity(nM)	%Rank	BindLevel
1	HLA-A*01:01	STPGHTIY	seq1	0.391	725.20	0.50	≤ SB
0	HLA-A*01:01	ASTPGHTII	seq1	0.071	23314.04	16.00	
7	HLA-A*01:01	IIYEAVCLH	seq1	0.060	26099.51	32.00	
13	HLA-A*01:01	CLHNDRTTI	seq1	0.042	31679.05	50.00	
6	HLA-A*01:01	TIIYEAVCL	seq1	0.040	32468.51	50.00	
4	HLA-A*01:01	GHTIIYEAV	seq1	0.039	32694.11	50.00	
5	HLA-A*01:01	HTIIYEAVC	seq1	0.036	33954.85	50.00	
10	HLA-A*01:01	EAVCLHNDR	seq1	0.025	37945.11	50.00	
11	HLA-A*01:01	AVCLHNDRT	seq1	0.025	38110.50	50.00	
3	HLA-A*01:01	PGHTIIYEAA	seq1	0.019	40758.73	50.00	
14	HLA-A*01:01	LHNDRTIIP	seq1	0.017	41823.30	50.00	
2	HLA-A*01:01	TPGHTIYE	seq1	0.016	42222.03	50.00	
8	HLA-A*01:01	IYEAVCLHN	seq1	0.016	42224.78	50.00	
9	HLA-A*01:01	YEAVCLHN	seq1	0.014	43073.35	50.00	
12	HLA-A*01:01	VCLHNDRT	seq1	0.010	45041.77	50.00	

Protein seq1. Allele HLA-A*01:01. Number of high binders 1. Number of weak binders 0. Number of peptides 15
Link to Allele Frequencies in Worldwide Populations [HLA-A*01:01](#)

Estructura de datos, generada por la clase NetMHC1_parse.

```
Array
(
    [0] => Array
        (
            [0] => result
            [1] => allele
            [2] => accuracy
            [3] => proteins
            [result] =>
                HLA-A01:01 : Estimated prediction accuracy 0.853 (using nearest neighbor HLA-A01:01)

                # Affinity Threshold for Strong binding peptides 50.000
                # Affinity Threshold for Weak binding peptides 500.000
                # Rank Threshold for Strong binding peptides 0.500
                # Rank Threshold for Weak binding peptides 2.000

                pos HLA peptide Identity 1-log50k(aff) Affinity(nM) %Rank BindLevel
-----
```

pos	HLA	peptide	Identity	1-log50k(aff)	Affinity(nM)	%Rank	BindLevel
1	HLA-A*01:01	STPGHTIY	seq1	0.391	725.20	0.50	SB
0	HLA-A*01:01	ASTPGHTII	seq1	0.071	23314.04	16.00	
7	HLA-A*01:01	IIYEAVCLH	seq1	0.060	26099.51	32.00	
13	HLA-A*01:01	CLHNDRTTI	seq1	0.042	31679.05	50.00	
6	HLA-A*01:01	TIIYEAVCL	seq1	0.040	32468.51	50.00	
4	HLA-A*01:01	GHTIIYEAV	seq1	0.039	32694.11	50.00	
5	HLA-A*01:01	HTIIYEAVC	seq1	0.036	33954.85	50.00	
10	HLA-A*01:01	EAVCLHNDR	seq1	0.025	37945.11	50.00	

11	HLA-A*01:01	AVCLHNDRT	seq1	0.025	38110.50	50.00
3	HLA-A*01:01	PGHTIIYEA	seq1	0.019	40758.73	50.00
14	HLA-A*01:01	LHNDRTTIP	seq1	0.017	41823.30	50.00
2	HLA-A*01:01	TPGHTIIYE	seq1	0.016	42222.03	50.00
8	HLA-A*01:01	IYEAVCLHN	seq1	0.016	42224.78	50.00
9	HLA-A*01:01	YEAVCLHND	seq1	0.014	43073.35	50.00
12	HLA-A*01:01	VCLHNDRTT	seq1	0.010	45041.77	50.00

Protein seq1. Allele HLA-A*01:01. Number of high binders 1. Number of weak binders 0. Number of peptides 15

[Link to Allele Frequencies in Worldwide Populations](#)

```
[allele] => HLA-A01:01
[accuracy] => 0.853
[proteins] => Array
(
    [0] => Array
        (
            [0] => full_analysis
            [1] => protein
            [2] => highBinders
            [3] => weakBinders
            [4] => peptidesTable
            [full_analysis] => Array
                (
                    [0] =>
                        Identity 1-log50k(aff) Affinity(nM) %Rank BindLevel
                    [1] =>
                        Identity 1-log50k(aff) Affinity(nM) %Rank BindLevel
                )
        )
)
pos HLA peptide

```

pos	HLA	peptide	Identity	1-log50k(aff)	Affinity(nM)	%Rank	BindLevel
1	HLA-A*01:01	STPGHTIIY	seq1	0.391	725.20	0.50	SB
0	HLA-A*01:01	ASTPGHTII	seq1	0.071	23314.04	16.00	
7	HLA-A*01:01	IIYEAVCLH	seq1	0.060	26099.51	32.00	
13	HLA-A*01:01	CLHNDRTTI	seq1	0.042	31679.05	50.00	
6	HLA-A*01:01	TIIYEAVCL	seq1	0.040	32468.51	50.00	
4	HLA-A*01:01	GHTIIYEAV	seq1	0.039	32694.11	50.00	
5	HLA-A*01:01	HTIIYEAVC	seq1	0.036	33954.85	50.00	
10	HLA-A*01:01	EAVCLHNDR	seq1	0.025	37945.11	50.00	
11	HLA-A*01:01	AVCLHNDRT	seq1	0.025	38110.50	50.00	
3	HLA-A*01:01	PGHTIIYEA	seq1	0.019	40758.73	50.00	
14	HLA-A*01:01	LHNDRTTIP	seq1	0.017	41823.30	50.00	
2	HLA-A*01:01	TPGHTIIYE	seq1	0.016	42222.03	50.00	
8	HLA-A*01:01	IYEAVCLHN	seq1	0.016	42224.78	50.00	
9	HLA-A*01:01	YEAVCLHND	seq1	0.014	43073.35	50.00	
12	HLA-A*01:01	VCLHNDRTT	seq1	0.010	45041.77	50.00	

[2] =>

Protein seq1. Allele HLA-A*01:01. Number of high binders 1. Number of weak binders 0. Number of peptides 15

[Link to Allele Frequencies in Worldwide Populations](#)

```
)
[protein] => seq1
[peptidesTable] => Array
(
    [0] => Array
        (
            [0] => pos
            [1] => HLA
            [2] => peptide
            [3] => Identity
        )
)
```

```

[4] => 1-log50k(aff)
[5] => Affinity(nM)
[6] => %Rank
[7] => BindLevel
)
[1] => Array
(
    [0] => Array
    (
        [0] => 1
        [1] => HLA-A*01:01
        [2] => STPGHTIIY
        [3] => seq1
        [4] => 0.391
        [5] => 725.20
        [6] => 0.50
        [7] => SB
    )
[1] => Array
(
    [0] => 0
    [1] => HLA-A*01:01
    [2] => ASTPGHTII
    [3] => seq1
    [4] => 0.071
    [5] => 23314.04
    [6] => 16.00
)
[2] => Array
(
    [0] => 7
    [1] => HLA-A*01:01
    [2] => IIYEAVCLH
    [3] => seq1
    [4] => 0.060
    [5] => 26099.51
    [6] => 32.00
)
[3] => Array
(
    [0] => 13
    [1] => HLA-A*01:01
    [2] => CLHNDRTTI
    [3] => seq1
    [4] => 0.042
    [5] => 31679.05
    [6] => 50.00
)
[4] => Array
(
    [0] => 6
    [1] => HLA-A*01:01
    [2] => TIIYEAVCL
    [3] => seq1
    [4] => 0.040
    [5] => 32468.51
    [6] => 50.00
)
[5] => Array

```

```

(
    [0] => 4
    [1] => HLA-A*01:01
    [2] => GHTIIYEAV
    [3] => seq1
    [4] => 0.039
    [5] => 32694.11
    [6] => 50.00
)
[6] => Array
(
    [0] => 5
    [1] => HLA-A*01:01
    [2] => HTIIYEAVC
    [3] => seq1
    [4] => 0.036
    [5] => 33954.85
    [6] => 50.00
)
[7] => Array
(
    [0] => 10
    [1] => HLA-A*01:01
    [2] => EAACLHNDR
    [3] => seq1
    [4] => 0.025
    [5] => 37945.11
    [6] => 50.00
)
[8] => Array
(
    [0] => 11
    [1] => HLA-A*01:01
    [2] => AVCLHNDRT
    [3] => seq1
    [4] => 0.025
    [5] => 38110.50
    [6] => 50.00
)
[9] => Array
(
    [0] => 3
    [1] => HLA-A*01:01
    [2] => PGHTIIYE
    [3] => seq1
    [4] => 0.019
    [5] => 40758.73
    [6] => 50.00
)
[10] => Array
(
    [0] => 14
    [1] => HLA-A*01:01
    [2] => LHNDRTTIP
    [3] => seq1
    [4] => 0.017
    [5] => 41823.30
)

```

```

[6] => 50.00
)
[11] => Array
(
    [0] => 2
    [1] => HLA-A*01:01
    [2] => TPGHTIIYE
    [3] => seq1
    [4] => 0.016
    [5] => 42222.03
    [6] => 50.00
)
[12] => Array
(
    [0] => 8
    [1] => HLA-A*01:01
    [2] => IYEAVCLHN
    [3] => seq1
    [4] => 0.016
    [5] => 42224.78
    [6] => 50.00
)
[13] => Array
(
    [0] => 9
    [1] => HLA-A*01:01
    [2] => YEAVCLHND
    [3] => seq1
    [4] => 0.014
    [5] => 43073.35
    [6] => 50.00
)
[14] => Array
(
    [0] => 12
    [1] => HLA-A*01:01
    [2] => VCLHNDRTT
    [3] => seq1
    [4] => 0.010
    [5] => 45041.77
    [6] => 50.00
)
)
)
)
[highBinders] => Array
(
    [0] => Array
    (
        [0] => 1
        [1] => HLA-A*01:01
        [2] => STPGHTIY
        [3] => seq1
        [4] => 0.391
        [5] => 725.20
        [6] => 0.50
        [7] => SB
    )
)
)
[weakBinders] => Array

```

()
()
()
()
()
()
()
()

Hoja de resultados de NetMHCIIpan con proteína Seq1 y alelo DRB5*02:05.

NetMHCIIpan Server - prediction results							
Technical University of Denmark							
# NetMHCIIpan version 3.0							
# Input is in FASTA format							
# Peptide length 15							
# Threshold for Strong binding peptides (IC50) 50.000 nM							
# Threshold for Weak binding peptides (IC50) 500.000 nM							
# Threshold for Strong binding peptides (%Rank) 0.5%							
# Threshold for Weak binding peptides (%Rank) 2%							
# Allele: DRB5*02:05							
pos	Allele	peptides	Identity	Pos	Core 1-log50(aff)	Affinity(nM)	%Rank
4	DRB5*02:05	GHTIYEAVCLHNDR	seq1	3	IYEAVVLH	0.698	26.25
5	DRB5*02:05	HTIIYEAVCLHNDRT	seq1	2	IYEAVVLH	0.574	33.87
3	DRB5*02:05	PGHTIYEAVCLHNN	seq1	4	IYEAVVLH	0.660	39.51
2	DRB5*02:05	TPGHIIYEAVCLHN	seq1	5	IYEAVVLH	0.658	40.40
6	DRB5*02:05	TIIYEAVCLHNDRTT	seq1	1	IYEAVVLH	0.640	48.99
1	DRB5*02:05	STPGHIIYEAVCLHNN	seq1	6	IYEAVVLH	0.620	61.18
7	DRB5*02:05	TCGHTIYEAVCLHNDRTT	seq1	0	IYEAVVLH	0.606	70.69
9	DRB5*02:05	IYEAVCLHNDRTTIP	seq1	3	AVCLHNDRT	0.404	288.24
0	DRB5*02:05	ASTPGHIIYEAVCL	seq1	6	TIIYEAVCL	0.381	811.12

Estructura de datos, generada por la clase NetMHCII_parse.

```

Array
(
    [0] => Array
        (
            [0] => result
            [1] => allele
            [2] => lines
            [3] => peptidesTable
            [4] => protein
            [5] => strongBinders
            [6] => weakBinders
            [result] => # Allele: DRB5*02:05
        )
    )
)

```

pos	Affinity (nM)	%Rank	Allele BindingLevel	peptide	Identity	Pos	Core 1-log50k(aff)		
4	26.25	32.00	SB	DRB5*02:05	GHTIIYEAVCLHNDR	seq1	3	IIYEAVCLH	0.698
5	33.87	50.00	SB	DRB5*02:05	HTIIYEAVCLHNDRT	seq1	2	IIYEAVCLH	0.674
3	39.51	50.00	SB	DRB5*02:05	PGHTIIYEAVCLHND	seq1	4	IIYEAVCLH	0.660

2		DRB5*02:05	TPGHTIIYEAVCLHN	seq1	5	IIYEAVCLH	0.658	
40.40	50.00	SB	DRB5*02:05	TIIYEAVCLHNDRTT	seq1	1	IIYEAVCLH	0.640
48.99	50.00	SB	DRB5*02:05	STPGHTIIYEAVCLH	seq1	6	IIYEAVCLH	0.620
61.18	50.00	WB	DRB5*02:05	IIYEAVCLHNDRTTI	seq1	0	IIYEAVCLH	0.606
70.69	50.00	WB	DRB5*02:05	IYEAVCLHNDRTTIP	seq1	3	AVCLHNDRT	0.494
238.24	50.00	WB	DRB5*02:05	ASTPGHTIIYEAVCL	seq1	6	TIIYEAVCL	0.381
811.12	50.00							
<hr/>								
Number of strong binders: 5 Number of weak binders: 3								
<hr/>								
<pre>[allele] => DRB5*02:05 [lines] => Array ([[0] => [pos => Allele %Rank => BindingLevel [1] => [[0] => [pos => Allele %Rank => BindingLevel [1] => [[0] => pos [1] => Allele [2] => peptide [3] => Identity [4] => Pos [5] => Core [6] => 1-log50k(aff) [7] => Affinity(nM) </pre>								
Affinity(nM)	%Rank	SB	peptide	Identity	Pos	Core	1-log50k(aff)	
26.25	32.00	SB	GHTIIYEAVCLHNDR	seq1	3	IIYEAVCLH	0.698	
33.87	50.00	SB	HTIIYEAVCLHNDRT	seq1	2	IIYEAVCLH	0.674	
39.51	50.00	SB	PGHTIIYEAVCLHND	seq1	4	IIYEAVCLH	0.660	
40.40	50.00	SB	TPGHTIIYEAVCLHN	seq1	5	IIYEAVCLH	0.658	
48.99	50.00	SB	TIIYEAVCLHNDRTT	seq1	1	IIYEAVCLH	0.640	
61.18	50.00	WB	STPGHTIIYEAVCLH	seq1	6	IIYEAVCLH	0.620	
70.69	50.00	WB	IIYEAVCLHNDRTTI	seq1	0	IIYEAVCLH	0.606	
238.24	50.00	WB	IYEAVCLHNDRTTIP	seq1	3	AVCLHNDRT	0.494	
811.12	50.00		ASTPGHTIIYEAVCL	seq1	6	TIIYEAVCL	0.381	
<hr/>								
Number of strong binders: 5 Number of weak binders: 3								
<hr/>								
<pre>) [peptidesTable] => Array ([[0] => Array ([[0] => pos [1] => Allele [2] => peptide [3] => Identity [4] => Pos [5] => Core [6] => 1-log50k(aff) [7] => Affinity(nM) </pre>								

```

[8] => %Rank
[9] => BindingLevel
)
[1] => Array
(
    [0] => Array
    (
        [0] => 4
        [1] => DRB5*02:05
        [2] => GHTIIYEAVCLHNDR
        [3] => seq1
        [4] => 3
        [5] => IIYEAVCLH
        [6] => 0.698
        [7] => 26.25
        [8] => 32.00
        [9] => SB
    )
    [1] => Array
    (
        [0] => 5
        [1] => DRB5*02:05
        [2] => HTIIYEAVCLHNDRT
        [3] => seq1
        [4] => 2
        [5] => IIYEAVCLH
        [6] => 0.674
        [7] => 33.87
        [8] => 50.00
        [9] => SB
    )
    [2] => Array
    (
        [0] => 3
        [1] => DRB5*02:05
        [2] => PGHTIIYEAVCLHND
        [3] => seq1
        [4] => 4
        [5] => IIYEAVCLH
        [6] => 0.660
        [7] => 39.51
        [8] => 50.00
        [9] => SB
    )
    [3] => Array
    (
        [0] => 2
        [1] => DRB5*02:05
        [2] => TPGHTIIYEAVCLHN
        [3] => seq1
        [4] => 5
        [5] => IIYEAVCLH
        [6] => 0.658
        [7] => 40.40
        [8] => 50.00
        [9] => SB
    )
    [4] => Array
    (

```

```

[0] => 6
[1] => DRB5*02:05
[2] => TIIYEAVCLHNDRTT
[3] => seq1
[4] => 1
[5] => IIYEAVCLH
[6] => 0.640
[7] => 48.99
[8] => 50.00
[9] => SB
)
[5] => Array
(
    [0] => 1
    [1] => DRB5*02:05
    [2] => STPGHTIYEAVCLH
    [3] => seq1
    [4] => 6
    [5] => IIYEAVCLH
    [6] => 0.620
    [7] => 61.18
    [8] => 50.00
    [9] => WB
)
[6] => Array
(
    [0] => 7
    [1] => DRB5*02:05
    [2] => IIYEAVCLHNDRTTI
    [3] => seq1
    [4] => 0
    [5] => IIYEAVCLH
    [6] => 0.606
    [7] => 70.69
    [8] => 50.00
    [9] => WB
)
[7] => Array
(
    [0] => 8
    [1] => DRB5*02:05
    [2] => IYEAVCLHNDRTTIP
    [3] => seq1
    [4] => 3
    [5] => AVCLHNDRT
    [6] => 0.494
    [7] => 238.24
    [8] => 50.00
    [9] => WB
)
[8] => Array
(
    [0] => 0
    [1] => DRB5*02:05
    [2] => ASTPGHTIYEAVCL
    [3] => seq1
    [4] => 6
    [5] => TIIYEAVCL
    [6] => 0.381
)

```



```

[7] => 40.40
[8] => 50.00
[9] => SB
)
[4] => Array
(
    [0] => 6
    [1] => DRB5*02:05
    [2] => TIIYEAVCLHNDRTT
    [3] => seq1
    [4] => 1
    [5] => IIYEAVCLH
    [6] => 0.640
    [7] => 48.99
    [8] => 50.00
    [9] => SB
)
)
[weakBinders] => Array
(
    [0] => Array
    (
        [0] => 1
        [1] => DRB5*02:05
        [2] => STPGHTIYEAVCLH
        [3] => seq1
        [4] => 6
        [5] => IIYEAVCLH
        [6] => 0.620
        [7] => 61.18
        [8] => 50.00
        [9] => WB
    )
    [1] => Array
    (
        [0] => 7
        [1] => DRB5*02:05
        [2] => IIYEAVCLHNDRTTI
        [3] => seq1
        [4] => 0
        [5] => IIYEAVCLH
        [6] => 0.606
        [7] => 70.69
        [8] => 50.00
        [9] => WB
    )
    [2] => Array
    (
        [0] => 8
        [1] => DRB5*02:05
        [2] => IYEAVCLHNDRTTIP
        [3] => seq1
        [4] => 3
        [5] => AVCLHNDRT
        [6] => 0.494
        [7] => 238.24
        [8] => 50.00
        [9] => WB
    )
)

```

```

        )
    )
)
```

Hoja de resultados de BepiPred con Seq1.

BepiPred 1.0b Server - prediction results							
Technical University of Denmark							
<hr/>							
<pre>##gff-version 2 ##source-version bepred-1.0b ##date 2014-06-09 ##Type Protein seq1 ##Protein seq1 cat seq.1.fsa ##ASTPGHTIIYEAVCLHNDRTTIP ##end-Protein # seqname source feature start end score N/A ? # -----</pre>							
seq1	bepred-1.0b	epitope	1	1	1.871	.	E
seq1	bepred-1.0b	epitope	2	2	1.750	.	E
seq1	bepred-1.0b	epitope	3	3	1.537	.	E
seq1	bepred-1.0b	epitope	4	4	1.008	.	E
seq1	bepred-1.0b	epitope	5	5	0.634	.	E
seq1	bepred-1.0b	epitope	6	6	0.445	.	E
seq1	bepred-1.0b	epitope	7	7	0.249	.	.
seq1	bepred-1.0b	epitope	8	8	-0.112	.	.
seq1	bepred-1.0b	epitope	9	9	-0.530	.	.
seq1	bepred-1.0b	epitope	10	10	-0.959	.	.
seq1	bepred-1.0b	epitope	11	11	-1.463	.	.
seq1	bepred-1.0b	epitope	12	12	-1.561	.	.
seq1	bepred-1.0b	epitope	13	13	-1.174	.	.
seq1	bepred-1.0b	epitope	14	14	-0.686	.	.
seq1	bepred-1.0b	epitope	15	15	-0.410	.	.
seq1	bepred-1.0b	epitope	16	16	-0.385	.	.
seq1	bepred-1.0b	epitope	17	17	-0.130	.	.
seq1	bepred-1.0b	epitope	18	18	-0.005	.	.
seq1	bepred-1.0b	epitope	19	19	0.248	.	.
seq1	bepred-1.0b	epitope	20	20	0.703	.	E
seq1	bepred-1.0b	epitope	21	21	0.892	.	E
seq1	bepred-1.0b	epitope	22	22	0.910	.	E
seq1	bepred-1.0b	epitope	23	23	0.684	.	E
# -----							

Estructura de datos, generada por la clase BepiPred_parse.

```
Array
(
    [0] => Array
        (
            [0] => result
            [1] => protein
            [2] => seq
            [3] => lines
            [4] => table
            [5] => epitopes
            [result] =>
##Type Protein seq1
##Protein seq1 cat seq.1.fsa
##ASTPGHTIIYEAVCLHNDRTTIP
##end-Protein
# seqname      source      feature      start      end      score      N/A      ?
# -----
```

```

seq1      bepipred-1.0b epitope      1      1      1.871 . E
seq1      bepipred-1.0b epitope      2      2      1.750 . E
seq1      bepipred-1.0b epitope      3      3      1.537 . E
seq1      bepipred-1.0b epitope      4      4      1.008 . E
seq1      bepipred-1.0b epitope      5      5      0.634 . E
seq1      bepipred-1.0b epitope      6      6      0.445 . E
seq1      bepipred-1.0b epitope      7      7      0.249 . .
seq1      bepipred-1.0b epitope      8      8      -0.112 . .
seq1      bepipred-1.0b epitope      9      9      -0.530 . .
seq1      bepipred-1.0b epitope     10     10      -0.959 . .
seq1      bepipred-1.0b epitope     11     11      -1.463 . .
seq1      bepipred-1.0b epitope     12     12      -1.561 . .
seq1      bepipred-1.0b epitope     13     13      -1.174 . .
seq1      bepipred-1.0b epitope     14     14      -0.686 . .
seq1      bepipred-1.0b epitope     15     15      -0.410 . .
seq1      bepipred-1.0b epitope     16     16      -0.385 . .
seq1      bepipred-1.0b epitope     17     17      -0.130 . .
seq1      bepipred-1.0b epitope     18     18      -0.005 . .
seq1      bepipred-1.0b epitope     19     19      0.248 . .
seq1      bepipred-1.0b epitope     20     20      0.703 . E
seq1      bepipred-1.0b epitope     21     21      0.892 . E
seq1      bepipred-1.0b epitope     22     22      0.910 . E
seq1      bepipred-1.0b epitope     23     23      0.684 . E
<b># -----</b>
[protein] => seq1
[seq] => ASTPGHTIIYEAVALHNDRTTIP
[lines] => Array
(
    [0] => seqname
    source
    feature
    start
    end
    score
N/A ??
    [1] =>
        seq1      bepipred-1.0b epitope      1      1      1.871 . E
seq1      bepipred-1.0b epitope      2      2      1.750 . E
seq1      bepipred-1.0b epitope      3      3      1.537 . E
seq1      bepipred-1.0b epitope      4      4      1.008 . E
seq1      bepipred-1.0b epitope      5      5      0.634 . E
seq1      bepipred-1.0b epitope      6      6      0.445 . E
seq1      bepipred-1.0b epitope      7      7      0.249 . .
seq1      bepipred-1.0b epitope      8      8      -0.112 . .
seq1      bepipred-1.0b epitope      9      9      -0.530 . .
seq1      bepipred-1.0b epitope     10     10      -0.959 . .
seq1      bepipred-1.0b epitope     11     11      -1.463 . .
seq1      bepipred-1.0b epitope     12     12      -1.561 . .
seq1      bepipred-1.0b epitope     13     13      -1.174 . .
seq1      bepipred-1.0b epitope     14     14      -0.686 . .
seq1      bepipred-1.0b epitope     15     15      -0.410 . .
seq1      bepipred-1.0b epitope     16     16      -0.385 . .
seq1      bepipred-1.0b epitope     17     17      -0.130 . .
seq1      bepipred-1.0b epitope     18     18      -0.005 . .
seq1      bepipred-1.0b epitope     19     19      0.248 . .
seq1      bepipred-1.0b epitope     20     20      0.703 . E
seq1      bepipred-1.0b epitope     21     21      0.892 . E
seq1      bepipred-1.0b epitope     22     22      0.910 . E
seq1      bepipred-1.0b epitope     23     23      0.684 . E
)





```

```

[0] => seqname
[1] => source
[2] => feature
[3] => start
[4] => end
[5] => score
[6] => N/A
[7] => ?

)
[1] => Array
(
    [0] => Array
    (
        [0] => seq1
        [1] => bepiped-1.0b
        [2] => epitope
        [3] => 1
        [4] => 1
        [5] => 1.871
        [6] => .
        [7] => E
    )
    [1] => Array
    (
        [0] => seq1
        [1] => bepiped-1.0b
        [2] => epitope
        [3] => 2
        [4] => 2
        [5] => 1.750
        [6] => .
        [7] => E
    )
    [2] => Array
    (
        [0] => seq1
        [1] => bepiped-1.0b
        [2] => epitope
        [3] => 3
        [4] => 3
        [5] => 1.537
        [6] => .
        [7] => E
    )
    [3] => Array
    (
        [0] => seq1
        [1] => bepiped-1.0b
        [2] => epitope
        [3] => 4
        [4] => 4
        [5] => 1.008
        [6] => .
        [7] => E
    )
    [4] => Array
    (

```

```

[0] => seq1
[1] => bepipred-1.0b
[2] => epitope
[3] => 5
[4] => 5
[5] => 0.634
[6] => .
[7] => E
)
[5] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 6
    [4] => 6
    [5] => 0.445
    [6] => .
    [7] => E
)
[6] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 7
    [4] => 7
    [5] => 0.249
    [6] => .
    [7] => .
)
[7] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 8
    [4] => 8
    [5] => -0.112
    [6] => .
    [7] => .
)
[8] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 9
    [4] => 9
    [5] => -0.530
    [6] => .
    [7] => .
)
[9] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 10
)

```

```

[4] => 10
[5] => -0.959
[6] => .
[7] => .
)
[10] => Array
(
    [0] => seq1
    [1] => bepiped-1.0b
    [2] => epitope
    [3] => 11
    [4] => 11
    [5] => -1.463
    [6] => .
    [7] => .
)
[11] => Array
(
    [0] => seq1
    [1] => bepiped-1.0b
    [2] => epitope
    [3] => 12
    [4] => 12
    [5] => -1.561
    [6] => .
    [7] => .
)
[12] => Array
(
    [0] => seq1
    [1] => bepiped-1.0b
    [2] => epitope
    [3] => 13
    [4] => 13
    [5] => -1.174
    [6] => .
    [7] => .
)
[13] => Array
(
    [0] => seq1
    [1] => bepiped-1.0b
    [2] => epitope
    [3] => 14
    [4] => 14
    [5] => -0.686
    [6] => .
    [7] => .
)
[14] => Array
(
    [0] => seq1
    [1] => bepiped-1.0b
    [2] => epitope
    [3] => 15
    [4] => 15
    [5] => -0.410
    [6] => .
    [7] => .
)

```

```

        )
[15] => Array
(
    [
        [0] => seq1
        [1] => bepipred-1.0b
        [2] => epitope
        [3] => 16
        [4] => 16
        [5] => -0.385
        [6] => .
        [7] => .
    ]
)
[16] => Array
(
    [
        [0] => seq1
        [1] => bepipred-1.0b
        [2] => epitope
        [3] => 17
        [4] => 17
        [5] => -0.130
        [6] => .
        [7] => .
    ]
)
[17] => Array
(
    [
        [0] => seq1
        [1] => bepipred-1.0b
        [2] => epitope
        [3] => 18
        [4] => 18
        [5] => -0.005
        [6] => .
        [7] => .
    ]
)
[18] => Array
(
    [
        [0] => seq1
        [1] => bepipred-1.0b
        [2] => epitope
        [3] => 19
        [4] => 19
        [5] => 0.248
        [6] => .
        [7] => .
    ]
)
[19] => Array
(
    [
        [0] => seq1
        [1] => bepipred-1.0b
        [2] => epitope
        [3] => 20
        [4] => 20
        [5] => 0.703
        [6] => .
        [7] => E
    ]
)
[20] => Array
(
    [
        [0] => seq1
    ]
)

```

```

[1] => bepipred-1.0b
[2] => epitope
[3] => 21
[4] => 21
[5] => 0.892
[6] => .
[7] => E
)
[21] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 22
    [4] => 22
    [5] => 0.910
    [6] => .
    [7] => E
)
[22] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 23
    [4] => 23
    [5] => 0.684
    [6] => .
    [7] => E
)
)
)
[epitopes] => Array
(
    [0] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 1
    [4] => 1
    [5] => 1.871
    [6] => .
    [7] => E
)
    [1] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 2
    [4] => 2
    [5] => 1.750
    [6] => .
    [7] => E
)
    [2] => Array
(
    [0] => seq1

```

```

[1] => bepipred-1.0b
[2] => epitope
[3] => 3
[4] => 3
[5] => 1.537
[6] => .
[7] => E
)
)
[3] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 4
    [4] => 4
    [5] => 1.008
    [6] => .
    [7] => E
)
)
[4] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 5
    [4] => 5
    [5] => 0.634
    [6] => .
    [7] => E
)
)
[5] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 6
    [4] => 6
    [5] => 0.445
    [6] => .
    [7] => E
)
)
[6] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 20
    [4] => 20
    [5] => 0.703
    [6] => .
    [7] => E
)
)
[7] => Array
(
    [0] => seq1
    [1] => bepipred-1.0b
    [2] => epitope
    [3] => 21
    [4] => 21
)
)

```

Quizás se considere que algunos datos dentro de las estructuras generadas están de más, lo cual puede ser cierto. Pero, el parseo fue diseñado de esa manera con tal de facilitar cualquier consulta de la hoja de resultados que se planee crear en un futuro. Es por eso que la estructura de datos considera todos los atributos que despliega la hoja de resultados.

1.8.5 Filtrado por localización subcelular

El servidor Cello nos indica la localización subcelular de una proteína. La localización subcelular es un aspecto de suma importancia para la elección de un buen candidato para vacuna. Es por ello, que se considera en REVXINE, la opción de realizar un filtrado, en la cual el usuario selecciona la localización subcelular de interés y REVXINE se encarga de enviar las proteínas con dicha localización subcelular a los demás servidores (NetMHCpan, NetMHCIIPan, BepiPred). Por lo tanto Cello se ejecuta antes que los demás servidores.

Para realizar el filtrado se programó en la clase Cello el siguiente método.

```
Fasta UpdateFasta(Fasta $fasta);
```

El método recibe de entrada un objeto del tipo Fasta ([Anexo 8](#)) y regresa este mismo pero actualizado. La actualización consiste en eliminar de la instancia, todas aquellas secuencias de aminoácidos que no se encuentran en la localización subcelular de interés.

El código en PHP es el siguiente:

```
public function UpdateFasta($fasta) {
    $this->location;
    if ($this->reliability == 'alta') {
        foreach ( $this->parse->results as $i => $result ) {
            $band = FALSE;
            foreach ( $this->location as $j => $location )
                if ($result ['best'] ['localization'] == $location)
                    $band = TRUE;

            if ($band == FALSE)
                $fasta->delete_seq ( $i );
        }
    } else {
        foreach ( $this->parse->results as $i => $result ) {
            $band = FALSE;
            foreach ( $this->location as $j => $location )
                foreach ( $result ['bests'] as $k => $best )
                    if ($best ['localization'] == $location)
                        $band = TRUE;

            if ($band == FALSE)
                $fasta->delete_seq ( $i );
        }
    }

    return $fasta;
}
```

El atributo actualizado es global para todos los servidores, entonces al ejecutar primero Cello, aseguramos que los otros servidores reciban las secuencias de aminoácidos que se encuentran en la localización subcelular de interés.

1.8.6 Creación de reportes

Con los arreglos multidimensionales generados por las clases parse de cada servidor, explicados anteriormente. Es posible realizar tablas o reportes, en formatos listos para publicar en artículos científicos.

En general el proceso es bastante simple, las actividades anteriores han sido elaboradas para llegar hasta este punto y poder generar reportes de la manera más sencilla posible.

Se creó una nueva clase llamada Reports ([Anexo 9](#)), a través de ella se generan los reportes y/o tablas para artículos científicos.

A continuación se muestran los reportes y la metodología para cada uno.

Reporte 1. Número de epítopos en MHCI y MHCII.

Tabla. Número de epítopos de células T (MHCI y II) estimados mediante NetMHC y NetMHCII.				
Proteína	Afinidad por MHC I (Tc)		Afinidad por MHC II (Th)	
	SB	WB	SB	WB
MAP0858	1	2	35	67
MAP1609C	2	10	67	126

Unión fuerte IC50s <50.000 nM y unión débil IC50s <500.000 nM para ambos MHC I y II.

Para crear esta tabla se cuentan los elementos de strongBinders y weakBinders en cada alelo para cada experimento de MHCI y MHCII, la sumatoria de estos elementos por proteína conforman el contenido de la tabla.

Reporte 2. Traslape de péptidos de MHC I y MHC II.

Tabla. Transposición epitopos de células T presentados por MHC I y MHC II.

Proteína	MHC I Epitopo Cel. T				MHC I Epitopo Cel. T			
	Secuencia	Posición	Allelo	Unión	Secuencia	Posición	Allelo	Unión
MAP0858	SVFAISCPL	159-167	HLA-A*02:01	SB	RSSVFAISCPLRHP	157-171	DRB1*01:01	SB
	FVLRSSVFA	154-162	HLA-A*02:01	WB	KSSKFVLRSSVFAIS	150-164	DRB1*01:01	SB
	VLRSSVFAI	155-163	HLA-A*02:01	WB	KSSKFVLRSSVFAIS	150-164	DRB1*01:01	SB
MAP1609C	FIYAGSLSA	182-190	HLA-A*02:01	SB	PDQFIYAGSLSALLD	179-193	DRB1*01:01	SB
	LLVGAAAAT	14-22	HLA-A*02:01	SB	GRRLLVGAAAATLIP	11-25	DRB1*01:01	SB
	GMGPSLIGL	197-205	HLA-A*02:01	WB	GMGPSLIGLAMGDAG	197-211	DRB1*01:01	WB
	GLAGGAATA	29-37	HLA-A*02:01	WB	LPGLIGLAGGAATAN	24-38	DRB1*01:01	SB
	ALLDPSQGM	190-198	HLA-A*02:01	WB	YAGSLSALLDPSQGM	184-198	DRB1*01:01	WB
	SMSGSSAMI	165-173	HLA-A*02:01	WB	AVGISMSGSSAMIILA	161-175	DRB1*01:01	SB
	YLQVPSAGM	49-57	HLA-A*02:01	WB	LPVEYLQVPSAGMGR	45-59	DRB1*01:01	SB
	GLPVEYLQV	44-52	HLA-A*02:01	WB	GLPVEYLQVPSAGMG	44-58	DRB1*01:01	SB
	RLLVGAAAA	13-21	HLA-A*02:01	WB	GRRLLVGAAAATLIP	11-25	DRB1*01:01	SB
	YQSGLSVIM	101-109	HLA-A*02:01	WB	FEWYYQSGLSVIMPV	97-111	DRB1*01:01	SB
	LTSELPSPYL	140-148	HLA-A*02:01	WB	ETFLTSELPSPYLASN	137-151	DRB1*01:01	SB
	SGLSVIMPV	103-111	HLA-A*02:01	WB	FEWYYQSGLSVIMPV	97-111	DRB1*01:01	SB

Para crear esta tabla es necesario realizar el experimento proteína por proteína. El experimento consiste en buscar en cada péptido de MHC I su coincidencia o traslape en péptidos de MHC II, solo la coincidencia con unión más fuerte es la que se toma en cuenta. Una vez encontrado el par de péptidos que coinciden o se traslanan, la información de cada uno se registra en un renglón de la tabla.

Reporte 3. Epítopos en MHCI.

Tabla. Epitopos Células T Citolíticas (9-aar) determinados por NetMHCI

Proteína	Secuencia	Alelo	Posición
MAP0858			
	SVFAISCPL	HLA-A*02:01	159
	FVLRSSVFA	HLA-A*02:01	154
	VLRSSVFAI	HLA-A*02:01	155
MAP1609C			
	FIYAGSLSA	HLA-A*02:01	182
	LLVGAAAAAV	HLA-A*02:01	14
	GMGPSLIGL	HLA-A*02:01	197
	GLAGGAATA	HLA-A*02:01	29
	ALLDPSQGM	HLA-A*02:01	190
	SMSGSSAMI	HLA-A*02:01	165
	YLQVPSAGM	HLA-A*02:01	49
	GLPVEYLQV	HLA-A*02:01	44
	RLLVGAAAAA	HLA-A*02:01	13
	YQSGLSVIM	HLA-A*02:01	101
	LTSELP SYL	HLA-A*02:01	140
	SGLSVIMPV	HLA-A*02:01	103

Para crear esta tabla se enlistan los epítopos encontrados de MHCI por cada proteína, desplegando la secuencia, Alelo, y su posición.

Con esta actividad se concluye uno de los propósitos fundamentales de REVXINE, construir automáticamente tablas o reportes en formatos listos para publicar en artículos científicos. Además, logramos reducir el tiempo y esfuerzo que los investigadores que trabajan en vacunas, gastan en analizar los resultados y armar estos formatos manualmente con herramientas de paquetería.

1.9 Evaluación o impacto económico, social o tecnológico

Las enfermedades son un problema sanitario muy costoso, el gobierno utiliza recursos públicos para proporcionar vacunas gratis a los ciudadanos a través del sector salud, es fundamental el desarrollo de vacunas menos costosas. Además, las enfermedades también se presentan en el ganado, en la agricultura y en la pesca. Por lo tanto, la salud se debe preservar en todos los sectores para un buen desarrollo económico y el bienestar social.

Esta herramienta da soporte a los investigadores que se especializan en el desarrollo de vacunas. Utilizando una técnica moderna llamada vacunología inversa, la cual consiste en utilizar sistemas de predicción inmunoinformáticos para encontrar buenos candidatos para vacunas.

El programa es libre, esto implica desde el punto de vista adquisitivo, que no tiene costo alguno. Además, es el único programa que cuenta con un módulo de generación automática de tablas en formatos científicos. Esta última característica de nuestro programa, representa un ahorro de tiempo y esfuerzo considerable para los investigadores. La cual no se consigue en otra aplicación.

1.10 Resultados Obtenidos

Se desarrolló una aplicación web que integra cuatro herramientas: Cello, utilizada para predicción de localización subcelular de proteínas, NetMHCI y NETMHCI para predicción de afinidad de acoplamiento a los complejos de histocompatibilidad clase I y II, además de Bepipred para la detección de epítopos de células B.

La selección de las herramientas a integrar se realizó en conjunto con los investigadores expertos del área de inmunología inversa.

Se logró desarrollar una biblioteca de clases que envía las peticiones a los servidores mediante la herramienta curl y se analizó los resultados, tanto exitosos como no exitosos, mediante el uso de expresiones regulares en PHP.

Se construyó una interfaz mediante un formulario único, para integrar los datos de petición necesarios para utilizar las cuatro herramientas seleccionadas.

Se diseñó un reporte de resumen por cada hoja de resultados y además tres tablas de reportes para resumir datos en conjunto, todo esto en una interfaz sencilla e intuitiva.

El producto final se nombró REVXINE (Reverse Vaccinology). Funciona como un portal hacia servidores externos, para la simplificación de uso y presentación de datos resumidos. Fue presentada y aprobada por el personal de CIBNOR y se ha implementado en sus servidores como un servicio público.

El programa cuenta con licencia GPL V2.0, y el repositorio oficial del programa se encuentra en <https://github.com/jlsalazar/REVXINE>.

1.11 Conclusiones y recomendaciones

Esta es la primera versión de REVXINE y sienta las bases para el desarrollo de herramientas computacionales mejoradas para la predicción de candidatos vacunales en colaboración interinstitucional entre el CIBNOR y el ITLP. El siguiente paso sería agregar nuevas aplicaciones para fortalecer los criterios de predicción y la generación de resultados complementarios.

Por lo tanto, me permito redactar algunas propuestas para trabajo futuro:

- Una vez terminado el sistema se llegó a la inquietud de poder ejecutar la comunicación con los servidores desde el lado del cliente, para ello se propone utilizar la librería curl sobre javascript, de esta forma se puede enviar desde el cliente múltiples peticiones desde un solo submit y esperar a que los servidores terminen la petición, con un mensaje amigable que ponga al corriente al cliente sobre el estado de su solicitud. Sea cual sea la respuesta por parte de los servidores, el javascript escrito en el cliente debe mandar la respuesta al servidor de REVXINE para que se someta a validación y posteriormente al análisis y muestra de reportes.
- También se plantea a consideración el agregar más servidores al sistema, por ahora la inquietud es agregar un servidor de predicción de células T llamado NetCTL, así también como un sistema de alineamiento de secuencias llamado BLAST. Esto para poder analizar y comparar en conjunto tanto predicción de epítopos de células B con epítopos de células T, y posteriormente BLAST para realizar filtros adicionales.

1.12 Bibliografía

- Conceptos biológicos. (<http://es.wikipedia.org/wiki/>).
- Inmunoinformática.
(http://www.pdg.cnb.uam.es/cursos/Complutense/Complutense2007/pages/07_Immuno/Immunoinformatics.pdf)
- Vacunología inversa. (<http://blogs.creamoselfuturo.com/biotecnologia/2011/07/11/vacunologia-inversa/>).
- Manual de expresiones regulares en PHP.
(http://www.mclibre.org/consultar/php/lecciones/php_expresiones_regulares.html).
- Christopher Murphy, Nicklas Persson (2009), HTML y CSS.
- Referencia cURL, oficial. (<http://curl.haxx.se/libcurl/php/>).
- Manual de referencia PHP. (<http://www.php.net/manual/es/langref.php>).
- Flanagan, David (2002). JavaScript: The Definitive Guide (4 ª Edición edición).
- NetMHC-3.0: accurate web accessible predictions of human, mouse and monkey MHC class I affinities for peptides of length 8-11. Lundegaard C, Lamberth K, Harndahl M, Buus S, Lund O, Nielsen M. Nucleic Acids Res. 1;36(Web Server issue):W509-12. 2008.
- NetMHCIIpan-3.0 , un método de predicción común-específica Pan MHC de clase II, incluyendo los tres isótipos humanos MHC de clase II, HLA-DR, HLA-DP y HLA-DQ .Karosiene E, Rasmussen H, Blicher T, Lund O, Buus S, y Nielsen M Inmunogenética, 2013.
- Yu CS, Lin CJ, Hwang JK: Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions. Protein Science 2004, 13:1402-1406.
- Método mejorado para la predicción de epítopos de células B lineales Jens Erik Pontoppidan Larsen, Ole Lund y Morten Nielsen Immunome Investigación 02:02, 2006 .

1.13 Anexos: planos, gráficas, prototipos y diagramas

1.13.1 Anexo 1. Pantalla principal de Cello v.2.5

MBC Molecular Bioinformatics Center
National Chiao Tung University

About CELLO

CELLO v.2.5: subCELLular LOcalization predictor

ORGANISMS	SEQUENCES
<input checked="" type="radio"/> Gram negative <input type="radio"/> Gram positive <input type="radio"/> Eukaryotes	<input type="radio"/> DNA <input checked="" type="radio"/> Protein

Paste the query sequences in FASTA format below

```
>1086005|genbank|outer membrane/Extracellular (Autotransporter)|major ring-forming surface protein precursor
MTKISDVQEKNFLRKKEKSSSLRNKRKFQPLIATTLAFSLASSFVNAADAGHAGQAPVNAEGITVTVNQANKTATVSGNN
GNATFTFTNGANTTVNGTADPAVTAPNIEVNIAINTVNIIFTVDGKPANQANQNLGAEGKPVVNLFDFGGIASSGTAKTFTL
NLGGAGNANALTNLNLGAGHATLNTNTNGSIASGGPVINVNKDATFNATFSGGATMTGNIIVTGNKTETSGTGTNNITF
DGPKQIPHNGSLIKDGTAVTGQADPATVLTGNISTYGGINNVTEFKGTMKDIIAGHATGQSLGMNVVTFKEQGVHYTGN
VIASGTTGVVNNTLNFGNATVDAUTNGONTLIIQNSGITSFVNNTNGVNNSPSTLTHATITPAAAGGDPANQATVFQGNIKSAYQ
GVNTLNFYNFNAKLEGPANKAAPAPAANITAATNNANHIVFTDGLVNMANTLSTLDQINTLVVNNTNNIVTNPIILTGNNV
VTNTPGWAGSNLTLLFQNNGTSSSTGGNAMQTLTNQVAYVGNIVANGGSVQAIIFSNTYWAPTNLKDILKEQAGGLNAAGAAGA
NARANAQAKSQQIQQGYLDDKFNGNSANATGNLTATNGGTATLVLRNTTTLANLPRQAAQYNVTVGGNNSSANIVLEAPVNA
SATITYGGYLGGNGTSNYVWNGSQNTSSVNLFIANADNRGPTLNGATGSSTLSDAFGGQFRNDLGAGKVLGVTYQNG
IQMSLSDKVNVLQGQNGLYSGSFMAFFKDAILAKIAKVDNAEFAQTQGIPLNVSLSVKGNGTSSPGSGGSFVNINITLEG
VAVGSITALTNKQATGTGNGMNTTSGVNLVLSKDSVLLGTIAGENQKGTLNMQLNQGAKLILQNSGAGTGGDVALNNLT
IASGNNNGNGNGAAVTFQGGSVSFQDNQANDYTALQNNNTVIDLATGGGSNNVPRTWFNLLTVQANSSNTTASDGQQA
SGLGGNNALFKVYVNADEHQNGNGAGGGGRGNATLNGQNSFNGSGLYGNIYSSDRVIVYQTQEQQHFCDRISPNNPRQWKSYGVR
YHGGTERAGNVAATVKNEGGQASVNFITVGSV1GFDFDAKLTAVKTNAYGKVEVNANINAGNSTPAPGLGSIPLGG
TGGTSSGNGTGGSQDQANAQDYTTYFISQAVANTSEANQLATATALASNYYLYLANIDSLNKRMGEELRSNPRSNGFWMRM
```

Or upload from file: No se eligió archivo

If you use CELLO in your publications, please cite one of the following publications:
(1) Yu CS, Lin CJ, Hwang JK: Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions. *Protein Science* 2004, 13:1402-1406.
(2) Yu CS, Chen YC, Lu CH, Hwang JK: Prediction of protein subcellular localization. *Proteins: Structure, Function and Bioinformatics* 2006, 64:643-651.

Contact

1.13.2 Anexo 2. Pantalla principal de NetMHCpan v.2.8

CENTERFO RIBIOLOGI CALSEQU ENCEANA LYSIS CBS	EVENTS STAFF	NEWS CONTACT	RESEARCH GROUPS ABOUT CBS	CBS PREDICTION SERVERS INTERNAL	CBS DATA SETS CBS BIOINFORMATICS TOOLS	PUBLICATIONS CBS COURSES	EDUCATION OTHER BIOINFORMATICS LINKS
--	-----------------	-----------------	------------------------------	------------------------------------	---	-----------------------------	---

CBS >> CBS Prediction Servers >> NetMHCpan-2.8

NetMHCpan 2.8 Server

NetMHCpan server predicts binding of peptides to any known MHC molecule using artificial neural networks (ANNs). The method is trained on more than 150,000 quantitative binding data covering more than 150 different MHC molecules. Predictions can be made for HLA-A, B, C, E and G alleles, as well as for non-human primates, mouse, Cattle and pig. Further, the user can upload full length MHC protein sequences, and have the server predict MHC restricted peptides from any given protein of interest.

Version 2.8 has been retrained on extended data set including 10 prevalent HLA-C and 7 prevalent BoLA MHC-I molecules.

Predictions can be made for 8-14 mer peptides. Note, that all non 9mer predictions are made using approximations. Most HLA molecules have a strong preference for binding 9mers.

The prediction values are given in nM IC50 values and as %-Rank to a set of 200,000 random natural peptides. For alleles distant to the MHC molecules included in the training of the method, only the Rank score is provided.

The project is a collaboration between CBS, [IMMI](#), and [LIA](#).

View the [version history](#) of this server. All the previous versions are available on line, for comparison and reference.

Link to table (tab separated) describing the training data: [Training data table](#)

As of July 8th, the nomenclature for BoLA-I has been updated to follow IPD Release 1.3.

Instructions	Output format	Article abstract	Evaluation Data
SUBMISSION			
Type of input <input type="button" value="Fasta"/>			
Paste a single sequence or several sequences in FASTA format into the field below.			
<input type="text"/>			
or submit a file in FASTA format directly from your local disk: <input type="button" value="Seleccionar archivo"/> No se eligió archivo			
Peptide length [several lengths are possible]: <input type="button" value="8mer peptides"/> <input type="button" value="9mer peptides"/> <input checked="" type="button" value="10mer peptides"/> <input type="button" value="11mer peptides"/>			
Select species/loci <input type="button" value="HLA supertype representative"/> <input type="button" value="Select Allele [max 20 per submission]"/>			
<input type="button" value="HLA-A*01:01 (A1)"/> <input type="button" value="HLA-A*02:01 (A2)"/> <input type="button" value="HLA-A*03:01 (A3)"/> <input type="button" value="HLA-A*24:02 (A24)"/> <input type="button" value="HLA-A*26:01 (A26)"/> <input type="button" value="HLA-B*07:02 (B7)"/> <input type="button" value="HLA-B*08:01 (B8)"/> <input type="button" value="HLA-B*27:05 (B27)"/> <input type="button" value="HLA-B*39:01 (B39)"/> <input type="button" value="HLA-B*40:01 (B44)"/> <input type="button" value="HLA-B*58:01 (B58)"/> <input type="button" value="HLA-B*15:01 (B62)"/>			
or type allele names (ie HLA-A01:01) separated by commas (and no spaces). Max 20 alleles per submission: <input type="text"/>			
For list of allowed allele names click here List of MHC allele names .			
or paste a single full length MHC protein sequence in FASTA format into the field below: <input type="text"/>			
or submit a file containing a full length MHC protein sequence in FASTA format directly from your local disk: <input type="button" value="Seleccionar archivo"/> No se eligió archivo			
Threshold for strong binder: % Rank <input type="text" value="0.5"/> OR IC50 <input type="text" value="50"/> Threshold for weak binder: % Rank <input type="text" value="2"/> OR IC50 <input type="text" value="500"/>			
Sort by affinity <input type="checkbox"/>			
Include IC50 prediction value for all alleles (default is for white-listed alleles only) <input type="checkbox"/>			
Save prediction to xls file <input type="checkbox"/>			
<input type="button" value="Submit"/> <input type="button" value="Clearfields"/>			
RESTRICTIONS: <i>At most 5000 sequences per submission, each sequence not more than 20,000 amino acids and not less than 8 amino acids. Max 20 MHC alleles per submission.</i>			
CONFIDENTIALITY: <i>The sequences are kept confidential and will be deleted after processing</i>			
CITATIONS For publication of results, please cite:			
<ul style="list-style-type: none"> • NetMHCpan - MHC class I binding prediction beyond humans Ikka Hoot, Björn Peters, John Sidney, Lasse Eggens Pedersen, Ole Lund, Søren Buus, and Morten Nielsen PMID: 19002800 Full text • NetMHCpan, a method for quantitative predictions of peptide binding to any HLA-A and -B locus protein of known sequence. Nielsen M, Lundsgaard C, Blicher T, Lamberth K, Hamdahl M, Justesen S, Roeder G, Peters B, Sette A, Lund O, Buus S PMID: 17726526 Full text 			

1.13.3 Anexo 3. Pantalla principal de NetMHCIIpan v3.0

CENTERFO RBIOLÓGI CALSEQU ENSAÑA LYSIS CBS	EVENTS STAFF	NEWS CONTACT	RESEARCH GROUPS ABOUT CBS	CBS PREDICTION SERVERS INTERNAL	CBS DATA SETS CBS BIOTINFORMATICS TOOLS	PUBLICATIONS CBS COURSES	EDUCATION OTHER BIOTINFORMATICS LINKS
--	---------------------	---------------------	--	--	--	------------------------------------	--

CBS > CBS Prediction Servers >> NetMHCIIpan 3.0

NetMHCIIpan 3.0 Server

NetMHCIIpan 3.0 server predicts binding of peptides to MHC class II molecules. The predictions are available for all three human MHC class II isotypes, HLA-DR, HLA-DP and HLA-DQ, as well as mouse molecules. NetMHCIIpan 3.0 server can produce predictions for peptides of 9 - 19 amino acids in length.

Two submission types are handled - the list of peptides or a protein sequence in FASTA format. The server provides a possibility for the user to choose MHC molecule in question from a long list of alleles or alternatively upload the MHC protein sequence of interest. For HLA-DP and HLA-DQ molecules alpha and beta chains can be chosen separately from the list. Also, for HLA-DP, HLA-DQ and mouse molecules, full length sequences of alpha and beta chains should be uploaded separately. For HLA-DR only beta chain sequence should be uploaded.

The prediction values are given in nM IC50 values and as % Rank to a set of 200,000 random natural peptides. The user has a choice of setting the threshold for defining strong and weak binders based on predicted affinity (IC50) or % Rank. Strong and weak binding peptides will be indicated in the output. The output can also be sorted based on predicted binding affinity as well as filtered on the user-specified thresholds.

The project is a collaboration between CBS, [IMI](#), and [UAB](#).

View the [version history](#) of this server. All the previous versions are available online, for comparison and reference.

Instructions	Output format	Article abstract	Data sets
SUBMISSION			
Type of input <input type="button" value="Fasta"/>			
Paste a single sequence or several sequences in FASTA format into the field below:			
<input type="text"/>			
or submit a file in FASTA format directly from your local disk: <input type="button" value="Seleccionar archivo"/> No se eligió archivo			
Peptide length <input type="text" value="15"/>			
Select species/loci <input type="button" value="DP"/>			
Select Allele [max. 20 per submission]			
<input type="checkbox"/> DPA1*0103 <input type="checkbox"/> DPB1*0101 <input type="checkbox"/> DPA1*0104 <input type="checkbox"/> DPB1*0201 <input type="checkbox"/> DPA1*0105 <input type="checkbox"/> DPB1*0202 <input type="checkbox"/> DPA1*0106 <input type="checkbox"/> DPB1*0301 <input type="checkbox"/> DPA1*0107 <input type="checkbox"/> DPB1*0401 <input type="checkbox"/> DPA1*0108 <input type="checkbox"/> DPB1*0402 <input type="checkbox"/> DPA1*0109 <input type="checkbox"/> DPB1*0501 <input type="checkbox"/> DPA1*0110 <input type="checkbox"/> DPB1*0601 <input type="checkbox"/> DPA1*0201 <input type="checkbox"/> DPB1*0801 <input type="checkbox"/> DPA1*0202 <input type="checkbox"/> DPB1*100001 <input type="checkbox"/> DPA1*0203 <input type="checkbox"/> DPB1*1001 <input type="checkbox"/> DPA1*0204 <input type="checkbox"/> DPB1*10101			
or type molecules names [e.g. DRB1_1121 or HLA-DQA10511-DQB10404] separated by commas (and no spaces). Max. 20 alleles per submission			
For the list of allowed molecule names click here: List of MHC molecule names .			
or paste a single full length protein sequence of MHC alpha chain in FASTA format into the field below:			
<input type="text"/>			
or submit a file containing a full length sequence of MHC alpha chain in FASTA format directly from your local disk: <input type="button" value="Seleccionar archivo"/> No se eligió archivo			
or paste a single full length protein sequence of MHC beta chain in FASTA format into the field below:			
<input type="text"/>			
or submit a file containing a full length sequence of MHC beta chain in FASTA format directly from your local disk: <input type="button" value="Seleccionar archivo"/> No se eligió archivo			
Threshold for strong binder (% Rank) <input type="text" value="0.5"/> Threshold for strong binder (IC50) <input type="text" value="50"/> Threshold for weak binder (% Rank) <input type="text" value="2"/> Threshold for weak binder (IC50) <input type="text" value="500"/>			
Filter output <input type="button" value="No"/>			
Use fast mode (recommended for large calculations) <input type="checkbox"/>			
Print only the strongest binding core <input type="checkbox"/>			
Sort output by affinity <input type="checkbox"/>			
Save predictions to xls file <input type="checkbox"/>			
<input type="button" value="Submit"/> <input type="button" value="Clear fields"/>			
Restrictions: Almost 5000 sequences per submission, each sequence not more than 20,000 amino acids and not less than 8 amino acids. Max 20 MHC alleles per submission.			
Confidentiality: <i>The sequences are kept confidential and will be deleted after processing</i>			
CITATIONS For publication of results, please cite: <ul style="list-style-type: none"> • NetMHCIIpan 3.0, a common pan-specific MHC class II prediction method including all three human MHC class II isotypes, HLA-DR, HLA-DP and HLA-DQ Karosiene E, Rasmussen M, Blicher T, Lund O, Buus S, and Nielsen M <i>Immunogenetics</i>, 2013 			

1.13.4 Anexo 4. Pantalla principal de BepiPred v.1.0

CENTERFOR BIOLOGICAL SEQUENCES CBS

EVENTS NEWS RESEARCH GROUPS CBS PREDICTION SERVERS CBS DATA SETS PUBLICATIONS EDUCATION

STAFF CONTACT ABOUT CBS INTERNAL CBS BIOINFORMATICS TOOLS CBS COURSES OTHER BIOINFORMATICS LINKS

[CBS >> CBS Prediction Servers >> BepiPred](#)

BepiPred 1.0 Server

BepiPred 1.0 server predicts the location of linear B-cell epitopes using a combination of a hidden Markov model and a propensity scale method.

INSTRUCTIONS | **Output format** | **Data sets** | **Article abstract**

SUBMISSION

Paste a single sequence or several sequences in [FASTA](#) format into the field below.

Submit a file in [FASTA](#) format directly from your local disk:

No se eligió archivo

Score threshold for epitope assignment

Restrictions:
At most 2000 sequences and 200,000 amino acids per submission, each sequence not less than 10 and not more than 6000 amino acids.

Confidentiality:
The sequences are kept confidential and will be deleted after processing.

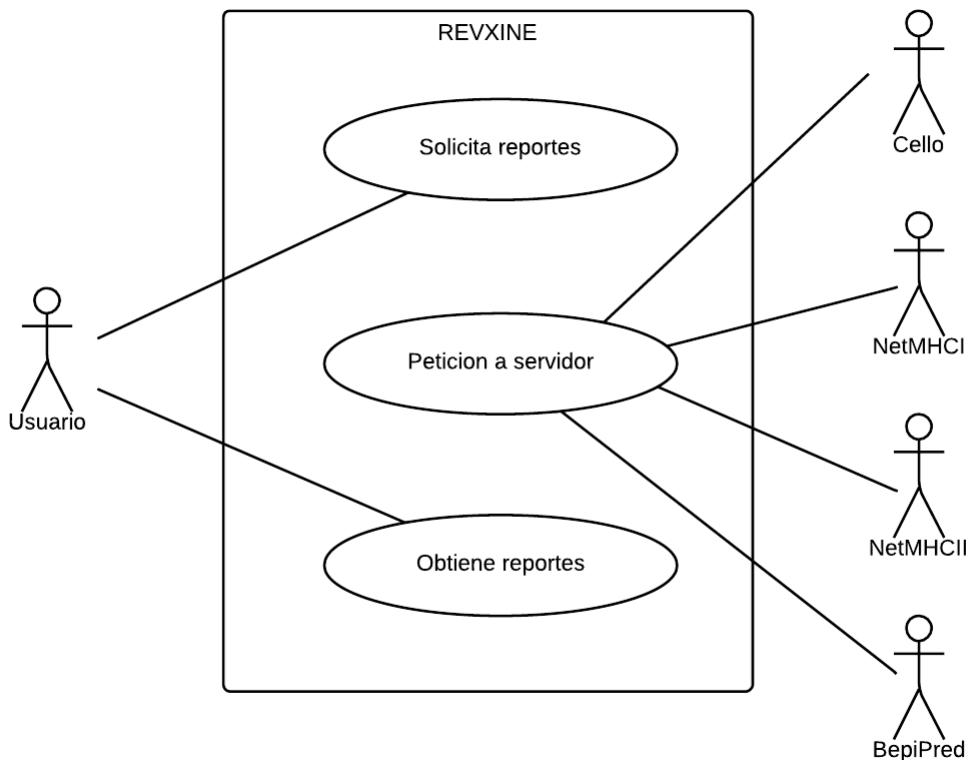
CITATIONS

For publication of results, please cite:

Improved method for predicting linear B-cell epitopes
Jens Erik Pontoppidan Larsen, Ole Lund and Morten Nielsen
Immunome Research 2:2, 2006.

View the [abstract](#).

1.13.5 Anexo 5. Diagrama de caso de uso.



Descripción de casos de uso

Caso de uso solicita reportes

Caso de Uso	Solicita reportes
Actores	Usuario
Descripción	El usuario envía una solicitud de análisis de secuencias en uno o más servidores inmunoinformáticos
Precondiciones	Ninguna
Flujo Básico de Datos	<ol style="list-style-type: none"> 1. El usuario introduce los datos; en particular: <ol style="list-style-type: none"> a. Ingresa un formato fasta las secuencias de aminoácidos. b. El usuario puede elegir utilizar múltiples servidores inmunoinformáticos para generar

	<p>sus reportes.</p> <ul style="list-style-type: none"> c. El usuario modifica los parámetros particulares de cada servidor. <ol style="list-style-type: none"> 2. Si el usuario acepta la operación. <ul style="list-style-type: none"> a. Si acepta con datos inválidos, excepción. b. Si acepta con datos válidos, petición al servidor.
Excepciones	<ol style="list-style-type: none"> 1. Error formato de secuencias FASTA inválido, termina el caso de uso.

Caso de uso Petición al servidor

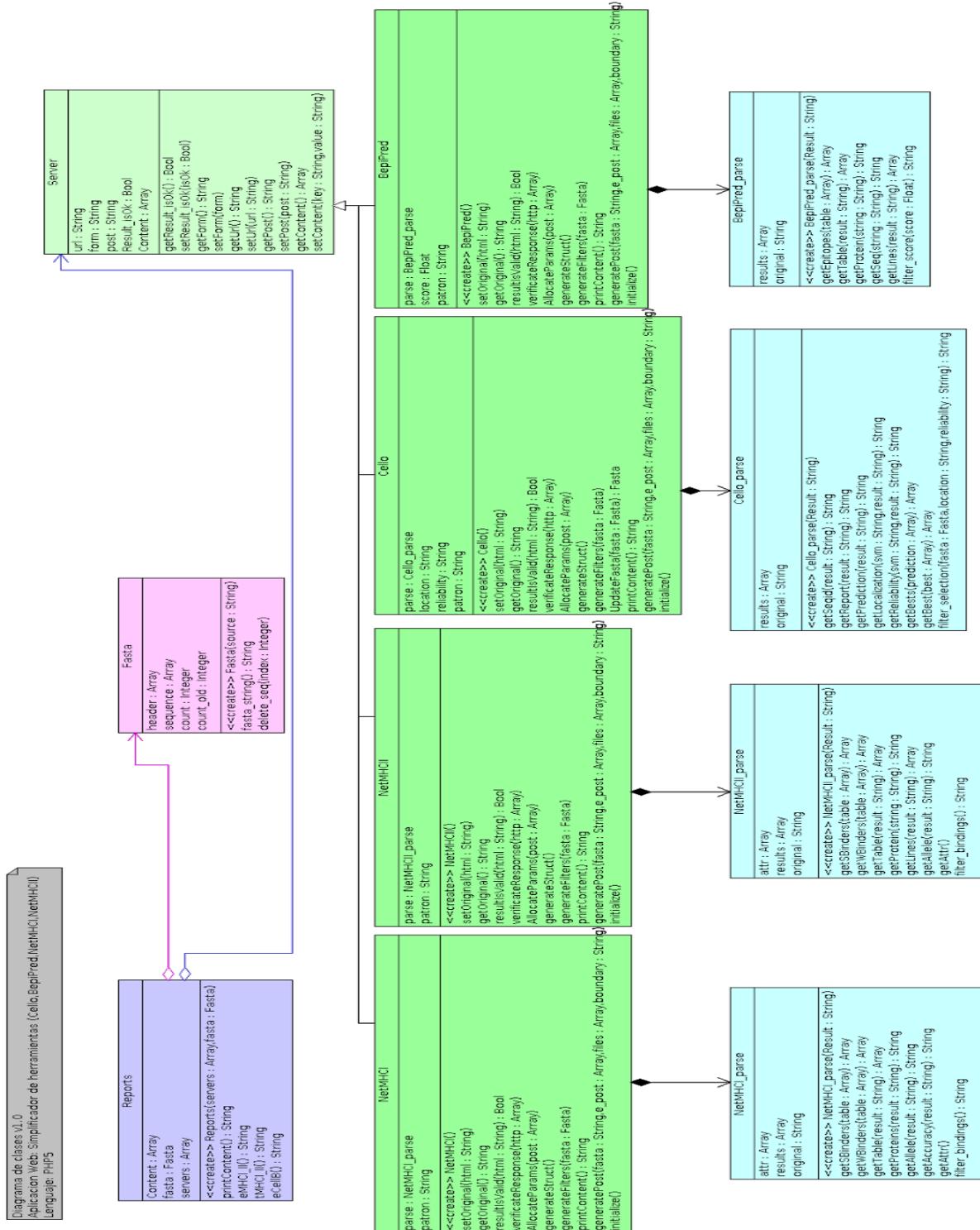
Caso de Uso	Petición al servidor
Actores	Cello, NetMHCI, NetMHCII, BepiPred
Descripción	El servidor procesa la petición del usuario y regresa un resultado.
Precondiciones	Formato FASTA correcto.
Flujo Básico de Datos	<ol style="list-style-type: none"> 3. REVXINE se comunica con el servidor; en particular. <ul style="list-style-type: none"> a. REVXINE envía la petición de usuario al servidor. b. REVXINE recibe respuesta. 4. Acción depende de respuesta. <ul style="list-style-type: none"> a. Si hay error en los parámetros, excepción. b. Si el servidor no pudo procesar la petición, excepción. c. Si la respuesta fue exitosa, obtiene reportes.
Excepciones	<ol style="list-style-type: none"> 2. Error parámetros incorrectos, termina caso de uso. 3. Error el servidor no pudo procesar la petición, termina caso de uso.

Caso de uso obtiene reportes

Caso de Uso	Obtiene reportes
Actores	Usuario

Descripción	Se despliega en formato de tablas los reportes.
Precondiciones	Ninguna
Flujo Básico de Datos	<ol style="list-style-type: none"> 1. Se pasa la respuesta de texto plano a una estructura de datos. 2. Se utiliza la estructura de datos para crear las tablas de reportes; en particular. <ol style="list-style-type: none"> a. Se genera tabla de resumen. b. Si hay más servidores analizados, se crean tablas de reportes analizando los datos en conjunto. c. Si no hay más servidores, termina caso de uso.
Excepciones	Ninguna

1.13.6 Anexo 6. Diagrama de clases



1.13.7 Anexo 7. Clases parse de cada servidor.

```
Cello_parse.php
<?php
include_once ('tools/tools.php');
include_once ('class/Fasta.php');
class Cello_parse {
    public $results = array ();
    public $original;
    function __construct($Result) {
        $original = str_replace ( array (
            "&nbsp;",
            "<br>",
            "</tr>",
            "<tr>"
        ), "", $Result );
        $results = divideResults ( $original, '<td COLSPAN=4>SeqID:' );
        foreach ( $results as $key => $result ) {
            $this->results [$key] = array (
                'result',
                'seqid',
                'report',
                'prediction',
                'bests',
                'best'
            );
            $this->results [$key] ['result'] = $result;
            $this->results [$key] ['seqid'] = $this->getSeqid ( $result );
            $this->results [$key] ['report'] = $this->getReport ( $result );
            $this->results [$key] ['prediction'] = $this->getPrediction ( $result );
            $this->results [$key] ['bests'] = $this->getBests ( $this->results [$key] ['prediction'] );
            $this->results [$key] ['best'] = $this->getBest ( $this->results [$key] ['bests'] );
        }
        $this->original = $original;
    }
    function getSeqid($result) {
        $i = strpos ( $result, '</td>', 0 );
        $seqid = substr ( $result, 20, $i - 20 );
        return $seqid;
    }
    function getReport($result) {
        $report = array (
            'Amino Acid Comp.' => array (
                'localization',
                'reliability'
            ),
            'N-peptide Comp.' => array (
                'localization',
                'reliability'
            ),
            'Partitioned seq. Comp.' => array (
                'localization',
                'reliability'
            ),
            'Physico-chemical Comp.' => array (
                'localization',
                'reliability'
            ),
            'Neighboring seq. Comp.' => array (
                'localization',
                'reliability'
            )
        );
    }
}
```

```

foreach ( $report as $svm => $value ) {
    $report[$svm]['localization'] = $this->getLocalization( $svm, $result );
    $report[$svm]['reliability'] = $this->getReliability( $svm, $result );
}
return $report;
}
function getPrediction($result) {
    $i = 0;
    $prediction = array ();
    $init = strpos ( $result, '<td></td>', 0 );
    while ( $init !== FALSE ) {
        $prediction[$i]['localization'] = $this->getLocalization( '<td></td>', $result );
        $prediction[$i]['reliability'] = $this->getReliability( '<td></td>', $result );
        $init = strpos ( $result, '<td></td>', $init + 4 );
        if ($init !== FALSE) {
            $result = substr ( $result, $init, strlen ( $result ) - $init );
            $init = 0;
        }
        $i++;
    }
    return $prediction;
}
function getLocalization($svm, $result) {
    $i = strpos ( $result, $svm, 0 );
    $i += 4;
    $i = strpos ( $result, '<td>', $i );
    $j = strpos ( $result, '</td>', $i );
    $i += 4;
    $seqid = trim ( substr ( $result, $i, $j - $i ) );
    return $seqid;
}
function getReliability($svm, $result) {
    $i = strpos ( $result, $svm, 0 );
    $i += 4;
    $i = strpos ( $result, '<td>', $i );
    $i = strpos ( $result, '</td>', $i );
    $i = strpos ( $result, '<td>', $i );
    $j = strpos ( $result, '</td>', $i );
    $i += 4;
    $seqid = substr ( $result, $i, $j - $i );
    return $seqid;
}
function printResults() {
    $print = '';
    print_r ( $this->results );
}
function getBests($prediction) {
    $bests = array ();
    $j = 0;
    foreach ( $prediction as $i => $value ) {
        if (strpos ( $value['reliability'], '*' , 0 ) !== FALSE) {
            $bests[$j] = $value;
            $bests[$j]['reliability'] = str_replace ( '*', '', $bests[$j]['reliability'] );
            $j++;
        }
    }
    return $bests;
}
function getBest($bests) {
    $best = array ();
    if (count ( $bests ) > 0) {
        $best = $bests[0];
        foreach ( $bests as $i => $value ) {
            if ( $best['reliability'] < $value['reliability'] )
                $best = $value;
        }
    }
    return $best;
}

```

```

}

    return $best;
}
function filter_selection($fasta, $location, $reliability) {
    $nombre_archivo_tmp = tempnam ( "tmp/", "FTA" );
    $pos = strpos ( $nombre_archivo_tmp, 'tmp/' , 0 );
    $nombre_archivo = substr ( $nombre_archivo_tmp, $pos, strlen ( $nombre_archivo_tmp ) - $pos ) .
'.fasta';

    savefile ( $nombre_archivo, $fasta->fasta_string () );
    unlink ( $nombre_archivo_tmp );

    if ($reliability == 'alta')
        $cadena = '<b>Tipo de b&uacute;queda:</b> Confiabilidad alta de localizaci&oacute;n en alguna de las
siguientes: ' . implode ( ",", $location );
    else
        $cadena = '<b>Tipo de b&uacute;queda:</b> Confiabilidad normal de localizaci&oacute;n en alguna de las
siguientes: ' . implode ( ",", $location );

    $cadena .= '<br>';
    $cadena .= '<b>Secuencias analizadas:</b> ' . $fasta->count_old . '<br>';
    $cadena .= '<b>Secuencias aceptadas:</b> ' . $fasta->count . '<br>';
    $cadena .= "<i><br>Se generó un archivo fasta con las secuencias aceptadas<br> <a href=" .
$nombre_archivo . " download='CelloFilter.fasta'>click aquí</a> para descargar.<br></i>";
    $cadena .= '<h3>Aceptados:</h3>';

    $cadena .= "<table>";
    $cadena .= '<tr>';
    $cadena .= "<td style='text-align:center;vertical-align:middle;width:500px;background-color: rgb(177,
177, 177);> Secuencia </td>";
    $cadena .= "<td colspan='2' style='text-align:center;vertical-align:middle;width:300px;background-
color: rgb(177, 177, 177);> Localizaci&oacute;n </td>";
    $cadena .= '</tr>';
    foreach ( $fasta->header as $key => $value ) {

        $cadena .= '<tr>';
        $cadena .= "<td rowspan='2' width='510px'> <div style='width:500px;height:auto;word-wrap: break-
word;'>" . '>' . $fasta->header [$key] ['full_line'] . '<br>' . $fasta->sequence [$key] . '</div></td>';
        $cadena .= "<td style='text-align:center;vertical-align:middle;background-color: rgb(204, 204,
204);> Confiabilidad alta</td>";
        $cadena .= "<td style='text-align:center;vertical-align:middle;background-color: rgb(204, 204,
204);> Confiabilidad normal</td>";
        $cadena .= '</tr>';

        $cadena .= '<tr>';
        $cadena .= "<td>";
        $cadena .= $this->results [$key] ['best'] ['localization'] . ' : ' . $this->results [$key] ['best'] [
'reliability'] . '<br>';
        $cadena .= '</td>';
        $cadena .= "<td>";
        foreach ( $this->results [$key] ['bests'] as $key2 => $value2 ) {
            $cadena .= $value2 ['localization'] . ' : ' . $value2 ['reliability'] . "<br>";
        }
        $cadena .= '</td>';
        $cadena .= '</tr>';
    }
    $cadena .= '</table>';
    return $cadena;
}
?>

```

```

NetMHC_parse.php
<?php
include_once ('tools/tools.php');
include_once ('class/Fasta.php');
class NetMHC_parse {
    public $attr = array (
        'peptide_len',
        'thrs',
        'thas',
        'thrw',
        'thaw'
    );
    public $results = array ();
    public $original;
    function __construct($Result) {
        $this->original = preg_replace ( "/<a(.*)a>/", "", str_replace ( "<=", "", $Result ) );
        $this->getAttr ();

        $results = divideResultsMHC ( $this->original, ': Estimated prediction accuracy' );
        foreach ( $results as $key => $result ) {
            $this->results [$key] = array (
                'result',
                'allele',
                'accuracy',
                'proteins'
            );
            $this->results [$key] ['result'] = $result;
            $this->results [$key] ['allele'] = $this->getAllele ( $result );
            $this->results [$key] ['accuracy'] = $this->getAccuracy ( $result );
            $this->results [$key] ['proteins'] = $this->getProteins ( $result );
            foreach ( $this->results [$key] ['proteins'] as $key2 => $result2 ) {
                $this->results [$key] ['proteins'] [$key2] ['protein'] = $this->getProtein ( $result2
['full_analisis'] [2] );
                $this->results [$key] ['proteins'] [$key2] ['peptidesTable'] = $this->getTable ( $result2
['full_analisis'] );
                $this->results [$key] ['proteins'] [$key2] ['highBinders'] = $this->getHBinders ( $this->results
[$key] ['proteins'] [$key2] ['peptidesTable'] );
                $this->results [$key] ['proteins'] [$key2] ['weakBinders'] = $this->getWBinders ( $this->results
[$key] ['proteins'] [$key2] ['peptidesTable'] );
            }
        }
    }
    function getHBinders($table) {
        $sb = array ();
        foreach ( $table [1] as $key => $value ) {
            $len = count ( $value );
            if ( $value [$len - 1] == 'SB' )
                $sb [$key] = $value;
        }
        $sb = array_values ( $sb );
        return $sb;
    }
    function getWBinders($table) {
        $wb = array ();
        foreach ( $table [1] as $key => $value ) {
            $len = count ( $value );
            if ( $value [$len - 1] == 'WB' )
                $wb [$key] = $value;
        }
        $wb = array_values ( $wb );
        return $wb;
    }
    function getTable($result) {
        $header = explode ( " ", preg_replace ( "/[ ]+/", " ", $result [0] ) );
        unset ( $header [0] );
        $header = array_values ( $header );
    }
}

```

```

$content = explode ( "\n", $result [1] );
unset ( $content [count ( $content ) - 1] );
unset ( $content [0] );
$content = array_values ( $content );

foreach ( $content as $key => $value ) {
    $content [$key] = explode ( " ", preg_replace ( "/[ ]+/", " ", $value ) );
    unset ( $content [$key] [0] );
    $content [$key] = array_values ( $content [$key] );
}

$content = updateIdentityMHCI ( $content );
return array (
    $header,
    $content
);
}
function getProtein($string) {
    $pos = strpos ( $string, 'Protein ', 0 );
    $pos2 = strpos ( $string, '.', $pos );
    $pos += 8;
    $protein = str_replace ( array (
        " ",
        "\n",
        "\r"
    ), "", substr ( $string, $pos, $pos2 - $pos ) );
    $pos = strpos ( $protein, "_" , 0 );
    if ($pos !== FALSE)
        $protein = substr ( $protein, 0, $pos );
    return $protein;
}
function getProteins($result) {
    $i = 0;
    $pos = 0;
    $protein = array ();

    while ( 1 ) {
        $pos = strpos ( $result, "-----"
-----", $pos + 1 );
        if ($pos === FALSE)
            break;
        $pos2 = strpos ( $result, "-----"
-----", $pos + 1 );
        if ($pos2 === FALSE)
            break;
        $pos2 = strpos ( $result, "-----"
-----", $pos2 + 1 );
        if ($pos2 === FALSE)
            break;
        $pos2 = strpos ( $result, "-----"
-----", $pos2 + 1 );
        if ($pos == FALSE)
            break;
        $pos2 += 83;
        $protein [$i] = array (
            'full_analisis',
            'protein',
            'highBinders',
            'weakBinders',
            'peptidesTable'
        );
        $protein [$i] ['full_analisis'] = explode ( "-----"
-----", substr ( $result, $pos, $pos2 - $pos ) );
    }
}

```

```

unset ( $protein [ $i ] [ 'full_analisis' ] [ 0 ] );
unset ( $protein [ $i ] [ 'full_analisis' ] [ 4 ] );
$protein [ $i ] [ 'full_analisis' ] = array_values ( $protein [ $i ] [ 'full_analisis' ] );
$pos = $pos2;
$i++;
}

return $protein;
}
function getAllele($result) {
$pos = 0;
$pos2 = strpos ( $result, ': Estimated prediction accuracy', 0 );
$allele = str_replace ( array (
" ",
"\n",
"\r"
), "", substr ( $result, $pos, $pos2 - $pos ) );
return $allele;
}
function getAccuracy($result) {
$pos = strpos ( $result, ': Estimated prediction accuracy', 0 );
$pos2 = strpos ( $result, ', 0' );
$pos += 31;
$accuracy = str_replace ( " ", "", substr ( $result, $pos, $pos2 - $pos ) );
return $accuracy;
}
function getAttr() {
$pos = strpos ( $this->original, '# Peptide length ', 0 );
$pos2 = strpos ( $this->original, "\n", $pos );
$pos += 17;
$this->attr [ 'peptide_len' ] = substr ( $this->original, $pos, $pos2 - $pos );

$pos = strpos ( $this->original, '# Affinity Threshold for Strong binding peptides ', 0 );
$pos2 = strpos ( $this->original, "\n", $pos );
$pos += 49;
$this->attr [ 'thas' ] = str_replace ( " ", "", substr ( $this->original, $pos, $pos2 - $pos ) );

$pos = strpos ( $this->original, '# Affinity Threshold for Weak binding peptides ', 0 );
$pos2 = strpos ( $this->original, "\n", $pos );
$pos += 47;
$this->attr [ 'thaw' ] = str_replace ( " ", "", substr ( $this->original, $pos, $pos2 - $pos ) );

$pos = strpos ( $this->original, '# Rank Threshold for Strong binding peptides ', 0 );
$pos2 = strpos ( $this->original, "\n", $pos );
$pos += 45;
$this->attr [ 'thrs' ] = str_replace ( " ", "", substr ( $this->original, $pos, $pos2 - $pos ) );

$pos = strpos ( $this->original, '# Rank Threshold for Weak binding peptides ', 0 );
$pos2 = strpos ( $this->original, "\n", $pos );
$pos += 43;
$this->attr [ 'thrw' ] = str_replace ( " ", "", substr ( $this->original, $pos, $pos2 - $pos ) );
}
function printResults() {
print_r ( $this->results );
}
function filter_bindings() {
$filter = '';

$filter .= "<b>Longitud del peptido: </b>" . $this->attr [ 'peptide_len' ] . "<br><br>";
$filter .= "<b>Umbral de afinidad para uni&ocutén fuerte: </b>" . $this->attr [ 'thas' ] . "<br>";
$filter .= "<b>Umbral de afinidad para uni&ocutén d&eacute;cutable: </b>" . $this->attr [ 'thaw' ] . "<br>";
$filter .= "<b>Umbral de rango para uni&ocutén fuerte: </b>" . $this->attr [ 'thrs' ] . "<br>";

```

```

$filter .= "<b>Umbral de rango para unicidad estimada de predicción:</b>" . $this->attr ['thrw'] . "<br>";
$filter .= '<br>';

foreach ( $this->results as $key => $result ) {
    $filter .= "<div> <h3>" . $result ['allele'] . " | " . " Precisión estimada de predicción : " . $result ['accuracy'];
    $filter .= "</h3></div>";
    foreach ( $result ['proteins'] as $key2 => $result2 ) {
        $filter .= "<table style='width:50%;'>";
        $filter .= "<tr><td id='mhc_header'>Proteína</td>
                <td id='mhc_header'>Peptidos</td>
                <td id='mhc_header'>Uniones fuertes</td>
                <td id='mhc_header'>Uniones débiles</td>
            </tr>";
        $filter .= "<td> . $result2 ['protein'] . "</td>";
        $filter .= "<td> . count ( $result2 ['peptidesTable'] [1] ) . "</td>";
        $filter .= "<td> . count ( $result2 ['highBinders'] ) . "</td>";
        $filter .= "<td> . count ( $result2 ['weakBinders'] ) . "</td>";
        $filter .= "</tr>";
        $filter .= "</table>";

        $filter .= "<table style='width:80%;'>

            $filter .= "<tr>";
            foreach ( $result2 ['peptidesTable'] [0] as $key3 => $result3 )
                $filter .= "<td id='mhc_header'>$result3</td>";
            $filter .= "</tr>";

            foreach ( $result2 ['highBinders'] as $key3 => $result3 ) {
                $filter .= "<tr>";
                foreach ( $result3 as $key4 => $result4 )
                    $filter .= "<td>$result4</td>";
                $filter .= "</tr>";
            }

            foreach ( $result2 ['weakBinders'] as $key3 => $result3 ) {
                $filter .= "<tr>";
                foreach ( $result3 as $key4 => $result4 )
                    $filter .= "<td>$result4</td>";
                $filter .= "</tr>";
            }

            $filter .= "</table>";
            $filter .= '<br>';
        }
    }
    $filter .= '<br>';
}
return $filter;
}

?>

```

```

NetMHCII_parse.php
<?php

include_once ('tools/tools.php');
include_once ('class/Fasta.php');
class NetMHCII_parse {
    public $attr = array (
        'peptide_len',
        'thrs',
        'thas',
        'thrw',
        'thaw'
    );
    public $results = array ();
    public $original;
    function __construct($Result) {
        $this->original = str_replace ( "<=", "", $Result );
        $this->getAttr ();
    }

    $results = divideResults ( $this->original, '# Allele:' );
    foreach ( $results as $key => $result ) {
        $this->results [$key] = array (
            'result',
            'allele',
            'lines',
            'peptidesTable',
            'protein',
            'strongBinders',
            'weakBinders'
        );
        $this->results [$key] ['result'] = $result;
        $this->results [$key] ['allele'] = $this->getAllele ( $result );
        $this->results [$key] ['lines'] = $this->getLines ( $this->results [$key] ['result'] );
        $this->results [$key] ['peptidesTable'] = $this->getTable ( $this->results [$key] ['lines'] );
        $this->results [$key] ['protein'] = $this->getProtein ( $this->results [$key] ['peptidesTable'] [1] );
    }
    $this->results  [$key]  ['strongBinders']  =  $this->getSBinders  (  $this->results  [$key]
['peptidesTable'] [1] );
    $this->results [$key] ['weakBinders'] = $this->getWBinders ( $this->results [$key] ['peptidesTable']
[1] );
}
function getSBinders($table) {
    $sb = array ();
    foreach ( $table as $key => $value ) {
        $len = count ( $value );
        if ( $value [$len - 1] == 'SB' )
            $sb [$key] = $value;
    }
    $sb = array_values ( $sb );
    return $sb;
}
function getWBinders($table) {
    $wb = array ();
    foreach ( $table as $key => $value ) {
        $len = count ( $value );
        if ( $value [$len - 1] == 'WB' )
            $wb [$key] = $value;
    }
    $wb = array_values ( $wb );
    return $wb;
}
function getTable($result) {
    $header = explode ( " ", preg_replace ( "/[ ]+/", " ", $result [0] ) );
    unset ( $header [0] );
}

```

```

$header = array_values ( $header );

$content = explode ( "\n", $result [1] );
unset ( $content [count ( $content ) - 1] );
unset ( $content [0] );
$content = array_values ( $content );

foreach ( $content as $key => $value ) {
    $content [$key] = explode ( " ", preg_replace ( "/[ ]+/", " ", $value ) );
    unset ( $content [$key] [0] );
    $content [$key] = array_values ( $content [$key] );
}

$content = updateIdentityMHCII ( $content );
return array (
    $header,
    $content
);
}

function getProtein($string) {
    return $string [0] [3];
}

function getLines($result) {
    $lines = array ();
    $lines = explode ( "-----", $result );
    unset ( $lines [0] );
    unset ( $lines [4] );
    $lines = array_values ( $lines );
    return $lines;
}

function getAllele($result) {
    $pos2 = strpos ( $result, "\n", 0 );
    $pos = 9;
    $allele = str_replace ( array (
        " ",
        "\n",
        "\r"
    ), "", substr ( $result, $pos, $pos2 - $pos ) );
    return $allele;
}

function getAttr() {
    $pos = strpos ( $this->original, '# Peptide length ', 0 );
    $pos2 = strpos ( $this->original, "\n", $pos );
    $pos += 17;
    $this->attr ['peptide_len'] = substr ( $this->original, $pos, $pos2 - $pos );

    $pos = strpos ( $this->original, '# Threshold for Strong binding peptides (IC50)', 0 );
    $pos2 = strpos ( $this->original, "\n", $pos );
    $pos += 47;
    $this->attr ['thas'] = str_replace ( array (
        " ",
        "nM"
    ), "", substr ( $this->original, $pos, $pos2 - $pos ) );

    $pos = strpos ( $this->original, '# Threshold for Weak binding peptides (IC50)', 0 );
    $pos2 = strpos ( $this->original, "\n", $pos );
    $pos += 45;
    $this->attr ['thaw'] = str_replace ( array (
        " ",
        "nM"
    ), "", substr ( $this->original, $pos, $pos2 - $pos ) );
}

```

```

$pos = strpos ( $this->original, '# Threshold for Strong binding peptides (%Rank)', 0 );
$pos2 = strpos ( $this->original, "\n", $pos );
$pos += 48;
$this->attr ['thrs'] = str_replace ( " ", "", substr ( $this->original, $pos, $pos2 - $pos ) );

$pos = strpos ( $this->original, '# Threshold for Weak binding peptides (%Rank)', 0 );
$pos2 = strpos ( $this->original, "\n", $pos );
$pos += 46;
$this->attr ['thrw'] = str_replace ( " ", "", substr ( $this->original, $pos, $pos2 - $pos ) );
}

function printResults() {
    print_r ( $this->results );
}

function filter_bindings() {
    $filter = '';

$filter .= "<b>Longitud del peptido: </b>" . $this->attr ['peptide_len'] . "<br><br>";
$filter .= "<b>Umbral de uni&oacute;n fuerte (IC50): </b>" . $this->attr ['thas'] . "<br>";
$filter .= "<b>Umbral de uni&oacute;n d&eacute;bil (IC50): </b>" . $this->attr ['thaw'] . "<br>";
$filter .= "<b>Umbral de uni&oacute;n fuerte (%Rank): </b>" . $this->attr ['thrs'] . "<br>";
$filter .= "<b>Umbral de uni&oacute;n d&eacute;bil (%Rank): </b>" . $this->attr ['thrw'] . "<br>";
$filter .= '<br>';

foreach ( $this->results as $key => $result ) {
    $filter .= "<div> <h3>" . $result ['allele'] . " | " . $result ['protein'];
    $filter .= "</h3></div>";

    $filter .= "<table style='width:50%;'>";
    $filter .= "<tr>
        <td id='mhc_header'>Peptidos</td>
        <td>
        <td id='mhc_header'>Uniones fuertes</td>
        <td>
        <td id='mhc_header'>Uniones debiles</td>
            </tr>";
    $filter .= "<tr>";
    $filter .= "<td> " . count ( $result ['peptidesTable'] [1] ) . "</td>";
    $filter .= "<td> " . count ( $result ['strongBinders'] ) . "</td>";
    $filter .= "<td> " . count ( $result ['weakBinders'] ) . "</td>";
    $filter .= "</tr>";
    $filter .= "</table>";

    $filter .= "<table style='width:100%;'>";
    $filter .= "<tr>";
    foreach ( $result ['peptidesTable'] [0] as $key2 => $result2 )
        $filter .= "<td id='mhc_header'>$result2</td>";
    $filter .= "</tr>";

    foreach ( $result ['strongBinders'] as $key2 => $result2 ) {
        $filter .= "<tr>";
        foreach ( $result2 as $key3 => $result3 ) {
            if ($key3 == 2) {
                $remplace = "<span class='bluetagS'>" . $result2 [5] . "</span>";
                $result3 = str_replace ( $result2 [5], $remplace, $result3 );
            }
            if ($key3 == 5)
                $result3 = "<span class='bluetagS'>" . $result3 . "</span>";
            $filter .= "<td>$result3</td>";
        }
        $filter .= "</tr>";
    }
}

```

```

        foreach ( $result ['weakBinders'] as $key2 => $result2 ) {
            $filter .= "<tr>";
            foreach ( $result2 as $key3 => $result3 ) {
                if ($key3 == 2) {
                    $remplace = "<span class='bluetagW'>" . $result2 [5] . "</span>";
                    $result3 = str_replace ( $result2 [5], $remplace, $result3 );
                }
                if ($key3 == 5)
                    $result3 = "<span class='bluetagW'>" . $result3 . "</span>";
                    $filter .= "<td>$result3</td>";
                }
                $filter .= "</tr>";
            }

            $filter .= "</table>";
        }
        return $filter;
    }
}
?>
BepiPred_parse.php
<?php
include_once ('tools/tools.php');
include_once ('class/Fasta.php');
class BepiPred_parse {
    public $results = array ();
    public $original;
    function __construct($Result) {
        $this->original = str_replace ( ". .", ".", $Result );

        $results = divideResults ( $this->original, '##Type Protein' );

        foreach ( $results as $key => $result ) {
            $this->results [$key] = array (
                'result',
                'protein',
                'seq',
                'lines',
                'table',
                'epitopes'
            );
            $this->results [$key] ['result'] = $result;
            $this->results [$key] ['protein'] = $this->getProtein ( $result );
            $this->results [$key] ['seq'] = $this->getSeq ( $result );
            $this->results [$key] ['lines'] = $this->getLines ( $result );
            $this->results [$key] ['table'] = $this->getTable ( $this->results [$key] ['lines'] );
            $this->results [$key] ['epitopes'] = $this->getEpitopes ( $this->results [$key] ['table'] [1] );
        }
    }
    function getEpitopes($table) {
        $epitopes = array ();
        foreach ( $table as $key => $value ) {
            $len = count ( $value );
            if ($value [$len - 1] == 'E')
                $epitopes [$key] = $value;
        }
        $epitopes = array_values ( $epitopes );
        return $epitopes;
    }
    function getTable($result) {
        $header = explode ( " ", preg_replace ( "/[ ]+/", " ", $result [0] ) );
        unset ( $header [0] );
        $header = array_values ( $header );

```

```

$content = explode ( "\n", $result [1] );
unset ( $content [count ( $content ) - 1] );
unset ( $content [0] );
$content = array_values ( $content );

foreach ( $content as $key => $value ) {
    $content [$key] = explode ( " ", preg_replace ( "/[ ]+/", " ", $value ) );
}

return array (
    $header,
    $content
);
}

function getProtein($string) {
    $pos2 = strpos ( $string, "\n", 0 );
    $pos = 14;

    $protein = preg_replace ( "/[ ]+/", " ", substr ( $string, $pos, $pos2 - $pos ) );

    $pos = strpos ( $protein, "_" );
    if ($pos !== FALSE)
        $protein = substr ( $protein, 0, $pos );
    return $protein;
}

function getSeq($string) {
    $pos = strpos ( $string, "\n", 1 );
    $pos = strpos ( $string, "\n", $pos + 1 );
    $pos2 = strpos ( $string, "#end-Protein", 0 );
    return str_replace ( array (
        "#",
        ""
    ), "", substr ( $string, $pos, $pos2 - $pos ) );
}

function getLines($result) {
    $pos = strpos ( $result, "#end-Protein", 1 );
    $pos = strpos ( $result, "\n", $pos );
    $pos2 = strpos ( $result, "#", $pos );
    $pos2 = strpos ( $result, "#", $pos2 + 1 );
    $pos2 = strpos ( $result, "#", $pos2 + 1 );
    if ($pos2 === FALSE)
        $pos2 = strlen ( $result );

    $string = str_replace ( "<b>", "", substr ( $result, $pos, $pos2 - $pos ) );
    $string = str_replace ( "-----", "",
    $string );
    $lines = array ();
    $lines = explode ( "#", $string );
    unset ( $lines [0] );

    $lines = array_values ( $lines );
    return $lines;
}

function filter_score($score) {
    $filter = '';
    $filter .= "<b>Puntuaci&on de umbral para la asignaci&on topo : </b> . $score .";
    "<br><br>";
    $filter .= '<br>';
    foreach ( $this->results as $key => $result ) {

        $filter .= "<div> <h3>" . $result ['protein'];
        $filter .= "</h3></div>";
        $filter .= "<div><b> Epitopos :</b> " . count ( $result ['epitopes'] );

```

```

$filter .= "</div>";
$filter .= "<div><b> Secuencia (" . strlen ( str_replace ( array (
    "\n",
    "\r"
), "", $result ['seq'] ) ) . "aa):</b> " . $result ['seq'];
$filter .= "</div>";

$filter .= "<table style='width:70%;'>";
$filter .= "<tr>";
foreach ( $result ['table'] [0] as $key2 => $result2 )
    $filter .= "<td id='mhc_header'>$result2</td>";
$filter .= "</tr>";

foreach ( $result ['epitopes'] as $key2 => $result2 ) {
    $filter .= "<tr>";
    foreach ( $result2 as $key3 => $result3 )
        $filter .= "<td>$result3</td>";
    $filter .= "</tr>";
}
$filter .= "</table>";
}
return $filter;
}
?>

```

1.13.8 Anexo 8. Clase Fasta.

```
class fasta {
    public $header = array () // array(['full_line'],[param])
    public $sequence = array () // array()
    public $count = 0;
    public $count_old = 0;
    function __construct($source) {
        $source_array = explode ( "\n", $source );
        $index = - 1;

        foreach ( $source_array as $line ) {
            $new_sequence = strpos ( $line, ">" );
            $new_note = strpos ( $line, ";" );

            if ($new_sequence === false && $new_note === false) {
                $this->sequence [$index] .= $line;
            } else {
                $index++;

                $line = substr ( $line, 1 );

                $this->header [$index] = array (
                    "full_line" => $line,
                    "param" => explode ( "|", $line )
                );
            }

            $this->sequence [$index] = '';
        }
        $this->count_old = $this->count = $index + 1;
    }
    function fasta_string() {
        $content = '';
        foreach ( $this->header as $i => $line )
            $content = $content . '>' . $this->header [$i] ['full_line'] . "\n" . $this->sequence
[$i] . "\n";
        return $content;
    }
    function delete_seq($index) {
        unset ( $this->header [$index] );
        unset ( $this->sequence [$index] );

        $this->count--;
    }
}
```

1.13.9 Anexo 9. Clase Reports.

```
<?php
include_once ('tools/tools.php');
include_once ('class/Fasta.php');
include_once ('class/classes.php');
class Reports {
    private $servers;
    private $fasta;
    private $Content = array ();
    function __construct($servers, $fasta) {
        $this->servers = $servers;
        $this->fasta = $fasta;
        $this->Content [0] = (isset ( $servers ['NetMHCI'] ) && isset ( $servers ['NetMHCII'] )) ?
        $this->eMHCI_II () : 'No disponible.';
        ;
        $this->Content [1] = (isset ( $servers ['NetMHCI'] ) && isset ( $servers ['NetMHCII'] )) ?
        $this->tMHCI_II () : 'No disponible.';
        $this->Content [2] = (isset ( $servers ['NetMHCI'] )) ? $this->eMHCI () : 'No
disponible.';
    }
    function getResult_isOK() {
        return 0;
    }
    public function printContent() {
        $print = "<div id='content_menu'>";
        foreach ( $this->Content as $i => $object ) {
            switch ($i) {
                case 0 :
                    $title = "Numero de Ep&iacutetopos(MHCI y II)";
                    break;
                case 1 :
                    $title = "Traslape(MHCI y II)";
                    break;
                case 2 :
                    $title = "Ep&iacutetopos(MHCI)";
                    break;
                default :
                    $title = $i;
            }
            $print = $print . "<a href='#' onclick=\"showmenu('$i', 'Reports_content'," . count (
                $this->Content ) . ")\">" . $title . "</a> | ";
        }
        $print = $print . "</div>";

        $print = $print . "<div id='content'>";
        foreach ( $this->Content as $i => $object ) {
            $print = $print . "<div id='Reports_content$i' style='display: none;'> " . $object .
        "</div>";
        }
        $print = $print . "</div>";

        return $print;
    }
    function eMHCI_II() {
        $servers = $this->servers;
        $MHCIparse = (isset ( $servers ['NetMHCI'] )) ? $servers ['NetMHCI']->parse : NULL;
    }
}
```

```

$MHCIIParse = (isset ( $servers ['NetMHCI'] )) ? $servers ['NetMHCI']->parse : NULL;

$report = '';
if ($MHCIIParse != NULL && $MHCIIParse != NULL) {
    $report .= "Tabla. N&uacute;mero de ep&iacutetopos de c&eacute;ctulas T (MHC I y II)
estimados mediante NetMHC y NetMHCII.<br>";

    $report .= "<div>";
    $report .= "<table style='width:50%;'>";
    $report .= "<tr>
        <td id='mhc_header'> Proteina</td>
        <td id='mhc_header'> Afinidad por MHC I (Tc)</td>
        <td id='mhc_header'> Afinidad por MHC II (Th)</td>
    </tr>";
    $report .= "<tr>
        <td id='mhc_header'> SB</td>
        <td id='mhc_header'> WB</td>
        <td id='mhc_header'> SB</td>
        <td id='mhc_header'> WB</td>
    </tr>";

    $proteinsMHCI = array ();
    foreach ( $MHCIIParse->results as $key => $result ) {
        foreach ( $result ['proteins'] as $key2 => $result2 ) {
            if (! isset ( $proteinsMHCI [ $result2 ['protein'] ] ))
                $proteinsMHCI [ $result2 ['protein'] ] = array (
                    'strongBinders' => 0,
                    'weakBinders' => 0
                );
            $proteinsMHCI [ $result2 ['protein'] ] ['strongBinders'] += count ( $result2
['highBinders'] );
            $proteinsMHCI [ $result2 ['protein'] ] ['weakBinders'] += count ( $result2
['weakBinders'] );
        }
    }

    $proteinsMHCIID = array ();
    foreach ( $MHCIIParse->results as $key => $result ) {
        if (! isset ( $proteinsMHCIID [ $result ['protein'] ] ))
            $proteinsMHCIID [ $result ['protein'] ] = array (
                'strongBinders' => 0,
                'weakBinders' => 0
            );
        $proteinsMHCIID [ $result ['protein'] ] ['strongBinders'] += count ( $result
['strongBinders'] );
        $proteinsMHCIID [ $result ['protein'] ] ['weakBinders'] += count ( $result
['weakBinders'] );
    }

    foreach ( $proteinsMHCI as $key => $result ) {
        $report .= "<tr>
            <td> $key</td>
            <td>
                $result[strongBinders]</td>
            <td>
                $result[weakBinders]</td>
        </tr>";
    }
}

```

```

[$key] ['strongBinders'] . " </td>
[$key] ['weakBinders'] . " </td>
}

$report .= "</table>";
$report .= "</div>";

$report .= "<div>";

if (($MHCIParse->attr ['thas'] == $MHCIIParse->attr ['thas']) && ($MHCIParse->attr ['thaw'] == $MHCIIParse->attr ['thaw']))
    $report .= "Uni&oacuten fuerte IC50s <" . $MHCIParse->attr ['thas'] . " nM y
uni&oacuten d&eacute;bil IC50s <" . $MHCIParse->attr ['thaw'] . " nM para ambos MHC I y II.";
else {
    $report .= "Uni&oacuten fuerte IC50s <" . $MHCIParse->attr ['thas'] . " nM y
uni&oacuten d&eacute;bil IC50s <" . $MHCIParse->attr ['thaw'] . " nM para MHC I.<br>";
    $report .= "Uni&oacuten fuerte IC50s <" . $MHCIIParse->attr ['thas'] . " nM y
uni&oacuten d&eacute;bil IC50s <" . $MHCIIParse->attr ['thaw'] . " nM para MHC II.<br>";
}
$report .= "</div>";
}

return $report;
}
function tMHCII() {
$servers = $this->servers;
$MHCIParse = (isset ( $servers ['NetMHCI'] )) ? $servers ['NetMHCI']->parse : NULL;
$MHCIIParse = (isset ( $servers ['NetMHCII'] )) ? $servers ['NetMHCII']->parse : NULL;

$report = '';
if ($MHCIParse != NULL && $MHCIIParse != NULL) {
    $report .= "Tabla. Transposici&oacuten epitopos de c&eacute;telulas T presentados por MHC I
y MHC II.<br>";

$report .= "<div>";
$report .= "<table style='width:95%;'>";
$report .= "<tr>
 96 | | | |
```



```

                $best = $result3;
            }
        }
    }
    if (count ( $best ) != 0) {
        $num = $result2 [0] + strlen ( $result2 [2] ) - 1;
        $num2 = $best [0] + strlen ( $best [2] ) - 1;

        $report .= "<tr><td></td>";
        $report .= "<td>" . $result2 [2] . "</td> <td>" . $result2 [0] . "-$num" .
"</td> <td>" . $result2 [1] . "</td> <td>" . $result2 [count ( $result2 ) - 1] . "</td>";
        $report .= "<td>" . str_replace ( $result2 [2], "<u>" . $result2 [2] . "</u>",
$best [2] ) . "</td> <td>" . $best [0] . "-$num2" . "" . "</td> <td>" . $best [1] . "</td>
<td>" . $best [count ( $best ) - 1] . "</td>";
        $report .= "</tr>";
    }
}
$report
"<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>";
}

$report .= "</table>";
$report .= "</div>";
}

return $report;
}
function eMHCI() {
    $servers = $this->servers;
    $fasta = $this->fasta;
    $MHCIparse = (isset ( $servers ['NetMHCI'] )) ? $servers ['NetMHCI']->parse : NULL;

    $report = '';
    if ($MHCIparse != NULL) {
        $report .= "Tabla. Epitopos C&eacute;telulas T Citot&eacute;xicas (9-aar) determinados por
NetMHCI";

        $report .= "<div>";
        $report .= "<table style='width:40%;'>";
        $report .= "<tr>
Prote&iacutena</td>
                                         <td id='mhc_header'>
Secuencia</td>
                                         <td id='mhc_header'>
Alelo</td>
                                         <td id='mhc_header'>
Posici&oacute;n</td>
                                         <td id='mhc_header'>
                                         </tr>";

        $i = 0;
        $proteinsMHCI = array ();
        foreach ( $MHCIparse->results as $key => $result ) {
            foreach ( $result ['proteins'] as $key2 => $result2 ) {
                foreach ( $result2 ['highBinders'] as $key3 => $result3 ) {

```

```

        $proteinsMHCII [$i] = $result3;
        $i++;
    }
    foreach ( $result2 ['weakBinders'] as $key3 => $result3 ) {
        $proteinsMHCII [$i] = $result3;
        $i++;
    }
}

$proteins = array ();
foreach ( $proteinsMHCII as $key => $result ) {
    if (! isset ( $proteins [ $result [ 3 ] ] ))
        $proteins [ $result [ 3 ] ] = array ();
    $len = count ( $proteins [ $result [ 3 ] ] );
    $proteins [ $result [ 3 ] ] [ $len ] = $result;
}

foreach ( $proteins as $key => $result ) {
    $report .= "<tr><td>$key</td><td></td><td></td><td></td><td></td><td></td></tr>";
    foreach ( $result as $key2 => $result2 ) {
        $report
"<tr><td></td><td>$result2[2]</td><td>$result2[1]</td><td>$result2[0]</td></tr>";
        }
    }
    $report .= "</table>";
    $report .= "</div>";
}
return $report;
}
?>
```

2 Documentos administrativos

2. 1 Carta de presentación y convenio con la empresa



Subsecretaría de Educación Superior
Dirección General de Educación Superior Técnica
Instituto Tecnológico de La Paz

La Paz, B.C.S. a 12/FEBRERO/2014.
Departamento: GESTIÓN TEC. Y VINC.
No. de Oficio: DGT/071/2014

ASUNTO: PRESENTACIÓN DEL ESTUDIANTE
Y AGRADECIMIENTO.

C. DR. SERGIO HERNÁNDEZ VAZQUEZ,
DIRECTOR GENERAL DEL CENTRO DE INVESTIGACIONES BIOLÓGICAS
DEL NOROESTE, S.C.,
P R E S E N T E.

El Instituto Tecnológico de La Paz, tiene a bien presentar a sus finas atenciones al (la) C. JUAN LUIS SALAZAR MENDOZA con número de control 10310129 de la carrera de INGENIERÍA EN SISTEMAS COMPUTACIONALES, quien desea desarrollar el proyecto de Residencias Profesionales, denominado "HERRAMIENTA WEB PARA LA INTEGRACIÓN DE SISTEMAS DE PREDICCIÓN DE REGIONES DE UNIÓN AL COMPLEJO MAYOR DE HISTOCOMPATIBILIDAD CLASE II" cubriendo un total de 640 horas, en un período de cuatro a seis meses.

Es importante hacer de su conocimiento que todos los estudiantes que se encuentran inscritos en esta institución cuentan con un seguro contra accidentes personales con la empresa Buffet de Asesoría y Consultoría en Riesgos, S.C., según póliza No AE 1489 e inscripción en el IMSS.

Así mismo, hacemos patente nuestro sincero agradecimiento por su buena disposición y colaboración para que nuestros estudiantes, aún estando en proceso de formación, desarrollos un proyecto de trabajo profesional, donde puedan aplicar el conocimiento y el trabajo en el campo de acción en el que se desenvolverán como futuros profesionistas.

Al vernos favorecidos con su participación en nuestro objetivo, sólo nos resta manifestarle la seguridad de nuestra más atenta y distinguida consideración.

A T E N T A M E N T E

"Ciencia es Verdad, Técnica es Libertad"

Mirechu Benilde Alcedo Marquez
LIC. MIRECHU BENILDE ALCEDO MARQUEZ,
JEFA DEL DEPARTAMENTO DE GESTIÓN
TECNOLÓGICA Y VINCULACIÓN

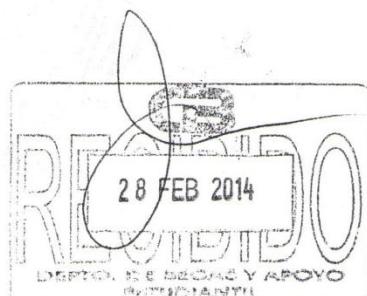
MBAM' asr*



Boulevard Forjadores de B.C.S. #4720, Col. 8 de oct. Tera. Sección C.P. 23080
La Paz, B.C.S. Comutador (612) 121-04-24, Fax: (612) 121-12-95
www.itlp.edu.mx



S.E.P.
S.E.S. D.G.E.S.T.
INSTITUTO TECNOLÓGICO DE LA PAZ
GESTIÓN TECNOLÓGICA Y VINCULACIÓN



ACUERDO DE TRABAJO QUE CELEBRAN POR UNA PARTE EL INSTITUTO TECNOLÓGICO DE LA PAZ, REPRESENTADO POR LA SUBDIRECCIÓN DE PLANEACIÓN Y VINCULACIÓN, POR ACUERDO DEL DIRECTOR, DR. OSCAR BAEZ SENTÍES, POR LA OTRA, CENTRO DE INVESTIGACIONES BIOLÓGICAS DEL NOROESTE, S.C., REPRESENTADO(A) EN ESTE ACTO POR EL (LA) DR. SERGIO HERNANDEZ VAZQUEZ, POR LA OTRA EL (LA) C. JUAN LUIS SALAZAR MENDOZA A QUIENES EN LO SUCESIVO SE LES DENOMINARÁ "EL TECNOLÓGICO", "EL CIBNOR" Y "EL ESTUDIANTE" RESPECTIVAMENTE, SUJETÁNDOSE A LAS SIGUIENTES DECLARACIONES Y CLÁUSULAS.

I.- DECLARACIONES:

Declara: "EL TECNOLÓGICO"

- 1.1 Ser una institución dedicada a impartir educación superior tecnológica de acuerdo a las necesidades que demanden los sectores productivos regionales y nacionales, adscrita a la Dirección General de Educación Superior Tecnológica dependiente de la Secretaría de Educación Pública y dotado de plena capacidad para suscribir el presente acuerdo de trabajo.
- 1.2 Que derivado de la Reforma de la Educación Superior Tecnológica, se incorporó a los planes de estudio de las carreras que se imparten en el Sistema Nacional de Educación Superior Tecnológica, el programa de Residencias Profesionales, y dado el carácter curricular de este programa se hace necesario fortalecer la vinculación entre la educación superior y los sectores productivos de bienes y servicios.
- 1.3 Que tiene establecido su domicilio en Boulevard Forjadores de Baja California Sur No. 4720 y que su R.F.C. es SEP-210905-778

Declara "EL CIBNOR"

- 1.4 Que su misión es: El Centro de Investigaciones Biológicas del Noroeste, S.C., es un Centro de Investigación, perteneciente al Sistema de Centros Públicos Conacyt, cuya misión es coadyuvar al bienestar de la sociedad mediante la realización de investigación científica, innovación tecnológica y formación de recursos humanos, en el manejo sustentable de los recursos naturales.
- 1.5 Que su domicilio se encuentra establecido en: Mar Bermejo No. 195 COL. PLAYA PALO DE SANTA RITA, en la Ciudad de La Paz, Baja California Sur y que su R.F.C. es CIB940530C73.

Declara "EL ESTUDIANTE"

- 1.6 Ser estudiante regular de "EL TECNOLÓGICO" inscrito en la carrera de **INGENIERÍA EN SISTEMAS COMPUTACIONALES**, con número de control **10310129**.
- 1.7 Haber acreditado como mínimo el **75%** del total de los créditos del Plan de Estudios de la carrera a que se hace referencia en la declaración 1.6.
- 1.8 Que desea realizar su Residencia Profesional en **EL CIBNOR**.
- 1.9 Que el proyecto denominado "**HERRAMIENTA WEB PARA LA INTEGRACIÓN DE SISTEMAS DE PREDICCIÓN DE REGIONES DE UNIÓN AL COMPLEJO MAYOR DE HISTOCOMPATIBILIDAD CLASE II**" es producto de la investigación preliminar realizada y que previo aval de la Academia de **SISTEMAS Y COMPUTACIÓN**, ha sido autorizado para su desarrollo por el Departamento de **SISTEMAS Y COMPUTACIÓN**.

II.- CLÁUSULAS:

PRIMERA. "EL CIBNOR", se compromete a brindar todas las facilidades necesarias tanto al asesor del proyecto como a "EL ESTUDIANTE" para el desarrollo del proyecto de Residencias Profesionales.

SEGUNDA. "EL ESTUDIANTE" se compromete a cumplir en tiempo y forma con todas las condiciones, características y requisitos que se mencionan en el proyecto a que se hace referencia en la declaración 1.9 y que es parte integrante del presente Acuerdo de Trabajo.

TERCERA. "EL ESTUDIANTE" se compromete a sujetarse al Manual de Procedimientos para la Planeación, Operación y Acreditación de las Residencias Profesionales vigente.

CUARTA. "EL ESTUDIANTE" cuenta con un seguro contra accidentes que pudieran ocurrir con motivo del desarrollo del proyecto que se menciona en la declaración 1.9 para lo cual "EL TECNOLÓGICO" a través de la Dirección General de Educación Superior Tecnológica, tiene contratado seguro contra accidentes personales con la empresa Buffet de Asesoría y Consultoría en Riesgos, S.C. según póliza No. AE1489.

Así mismo "EL TECNOLÓGICO" se compromete a verificar que "EL ESTUDIANTE" tenga actualizado sus servicios médicos.

QUINTA. Con motivo del desarrollo de algún proyecto de Residencias Profesionales y debido al carácter académico que tienen "EL ESTUDIANTE" acepta que su participación no originará ninguna relación laboral ni de seguridad social con "EL CIBNOR".

SEXTA. Los apoyos que "EL CIBNOR" pueda otorgar a "EL ESTUDIANTE" serán de acuerdo a su normatividad y disponibilidad presupuestal.

SÉPTIMA. "EL ESTUDIANTE" y "EL TECNOLÓGICO" se comprometen con "EL CIBNOR" a resguardar la confidencialidad de la información que surja con motivo de la implementación del presente acuerdo y que se aplique única y exclusivamente al desarrollo del proyecto a que se refiere la declaración 1.9.

Leído que fue el presente Acuerdo y enteradas las partes de su contenido, alcance y fuerza legal, se firma al margen y al calce en la ciudad de La Paz, Baja California Sur a los Doce días del mes de Febrero del año Dos mil catorce.

POR EL TECNOLÓGICO

M.C. ANTONIO HERNANDEZ SÁNCHEZ
SUBDIRECTOR DE PLANEACIÓN Y
VINCULACIÓN

EL CIBNOR

DR. ELENA SERVIERE ZARAGOZA
DIRECTORA DE ESTUDIOS DE POSGRADO

EL ESTUDIANTE

JUAN LUIS SALAZAR MENDOZA
NO. DE CONTROL 10310129
INGENIERÍA EN SISTEMAS COMPUTACIONALES

EL ASESOR EXTERNO

DR. CARLOS ELIUD ANGULO VALADEZ
INVESTIGADOR ASOCIADO C

2.2 Evaluación interna

	EVALUACIÓN INTERNA DEL PROYECTO DE RESIDENCIAS PROFESIONALES
Esta evaluación es realizada por el Asesor Interno	

NOMBRE DEL RESIDENTE: JUAN LUIS SALAZAR MENDOZA
 CARRERA: ING. EN SISTEMAS COMPUTACIONALES No. De control 10310129
 NOMBRE DE LA EMPRESA: CENTRO DE INVESTIGACIONES BIOLOGICAS DEL NOROESTE
 NOMBRE DEL PROYECTO: HERRAMIENTA WEB PARA LA INTEGRACIÓN DE SISTEMAS DE PREDICCIÓN DE REGIONES DE UNIÓN AL COMPLEJO MAYOR DE HISTOCOMPATIBILIDAD CLASE II.
 DEPARTAMENTO ASIGNADO: SISTEMAS Y COMPUTACIÓN
 PERÍODO DE LA RESIDENCIA: Del: 27 de Enero al 27 de Mayo del 2014
 FECHA DE LA EVALUACIÓN: 18 de Junio del 2014

INSTRUCCIONES: Marcar con una X para cada criterio de evaluación, su percepción sobre el desempeño del estudiante residente en el cumplimiento de los objetivos de su proyecto.

CRITERIO DE EVALUACIÓN	D	A	B	E	CÓDIGO
CUMPLIMIENTO DE LOS OBJETIVOS				X	
CALIDAD EN SUS ACTIVIDADES				X	D: DEFICIENTE
DISCIPLINA				X	A: ACEPABLE
COOPERACIÓN				X	B: BUENO
RESPONSABILIDAD				X	E: EXCELENTE
INICIATIVA				X	
SOCIABILIDAD				X	

EVALUACION (ESCALA 0 A 100 %) 100%

COMENTARIOS:


 DR. MARCO ANTONIO CASTRO LIERA
 ASESOR INTERNO
 (NOMBRE Y FIRMA)



ORIGINAL: DEPARTAMENTO ACADÉMICO
 COPIA: ASESOR INTERNO.
 COPIA: DEPTO. DE GESTIÓN TECN. Y VINCULACIÓN.
 ITLP-AC-PO-007-07

Rev. 0

2.3 Evaluación externa

	Nombre del documento: Evaluación del proyecto de residencias profesionales	Código:ITLP-AC-PO-007-07 Revisión: 0
	Referencia a la Norma ISO 9001:2008 7.5.1	Página 1 de 2

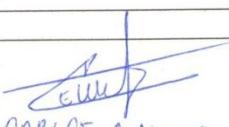
EVALUACIÓN DEL PROYECTO DE RESIDENCIAS PROFESIONALES

NOMBRE DEL RESIDENTE: JUAN LUIS SALAZAR MENDOZA
 CARRERA: INGENIERÍA EN SISTEMAS COMPUTACIONALES No. DE CONTROL 10310129
 NOMBRE DE LA EMPRESA: CENTRO DE INVESTIGACIONES BIOLÓGICAS DEL NOROESTE
 DEPARTAMENTO ASIGNADO: AGRICULTURA EN ZONAS ÁRIDAS
 PERÍODO DE LA RESIDENCIA: DEL 27 DE ENERO AL 27 DE MAYO DEL 2014.
 FECHA DE EVALUACIÓN: 17 DE JUNIO DEL 2014
 INSTRUCCIONES: Marcar con una X para cada criterio de evaluación, su percepción sobre el desempeño del estudiante residente en el cumplimiento de los objetivos de su proyecto.

CRITERIO DE EVALUACIÓN	D	A	B	E	CÓDIGO
CUMPLIMIENTO DE LOS OBJETIVOS				X	
CALIDAD EN SUS ACTIVIDADES				X	
DISCIPLINA				X	
COOPERACIÓN				X	
RESPONSABILIDAD				X	
INICIATIVA				X	
SOCIABILIDAD				X	

EVALUACION (ESCALA 0 A 100 %) 100%

COMENTARIOS: EL pISC. SALAZAR ES UN EXCELENTE PROFESIONAL, DEDICADO Y RESPONSABLE. COMENTAR MI FELICITACIÓN A LOS MAESTROS QUE LO FORMARON Y AGRADEZCO AL pISC. SALAZAR POR SU APOYO TRASCENDENTAL EN LA EJECUCIÓN DE ESTE PROYECTO DE RESIDENCIAS E INVESTIGACIÓN.


CARLOS ANGULO
 ASESOR EXTERNO
 (NOMBRE Y FIRMA)

