

**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS**

**FACULTAD 4**



**“IMPLEMENTACION DE LA INTERFAZ DE USUARIO DEL SISTEMA  
INTEGRAL PARA LA GESTION DE RECURSOS HUMANOS”**

**Trabajo de diploma para optar por el título de Ingeniería en Ciencias  
Informáticas**

**Autor(es):** Yiliexy Martinez Garcia

Hugo Leonardo Benítez García

**Tutor:** Ing. Yosvany Medina Hernández

**Consultante:** MSc. Julio C Días Vera

**Ciudad de La Habana, Cuba**

**Junio del 2008**

## DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

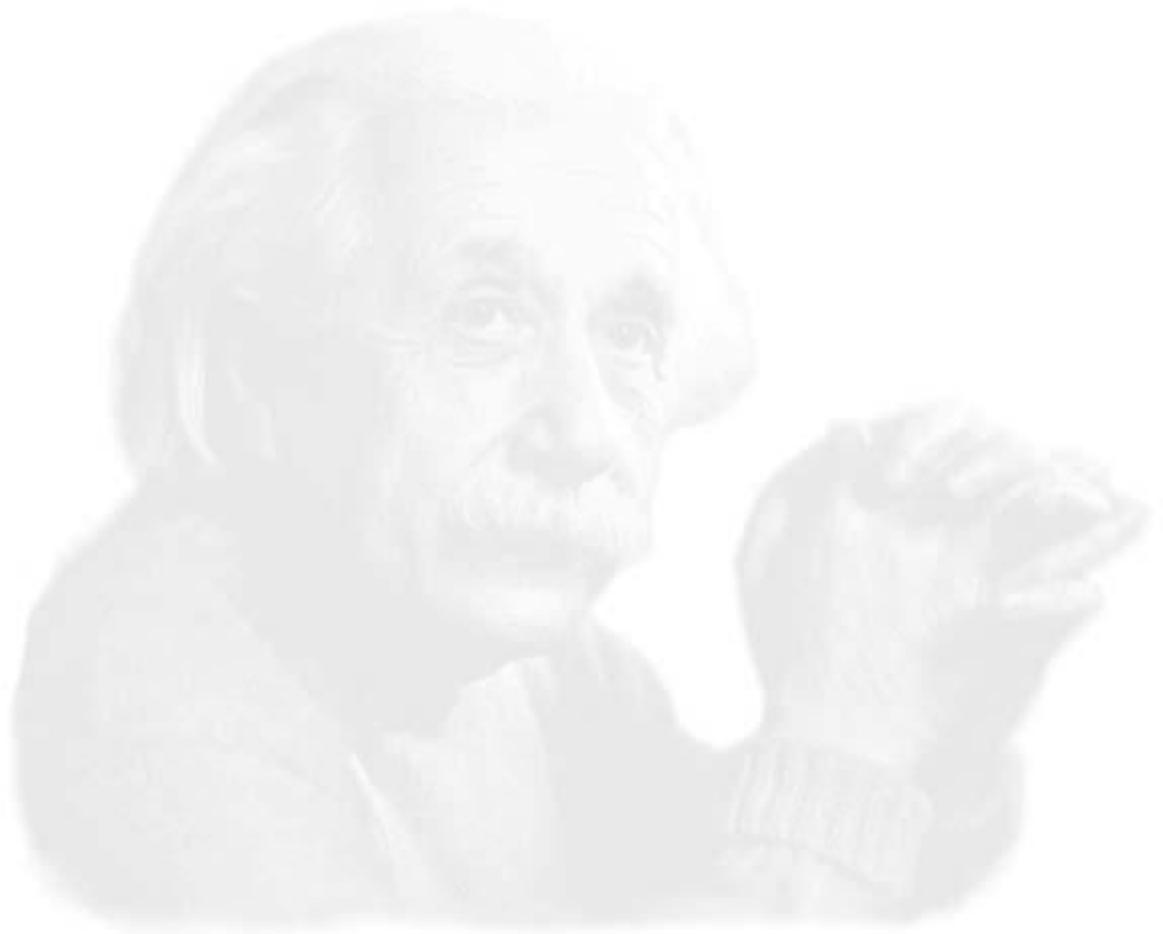
Yiliexy Martinez Garcia

---

Hugo Leonardo Benítez García

---

Ing. Yosvany Medina Hernández



*Comienza a manifestarse la madurez cuando sentimos que  
nuestra preocupación es mayor por los demás que por  
nosotros mismos.*

*Albert Einstein.*

## AGRADECIMIENTOS

*A ti Dios por darme la oportunidad de vivir y regalarme una familia tan maravillosa. Por enseñarme el camino correcto de la vida, guiándome y fortaleciéndome cada día con su espíritu.*

*A Fidel y a la **Revolución** cubana por permitirme estudiar en la mejor escuela del país  
Principalmente a mi adorada madre: **Ramona Martínez García** por ser la mejor, la que me dio la vida. Por su determinación, entrega y humildad que me ha enseñado tanto. Por ser la razón de mí existir. Gracias por todo mami, por apoyarme en todas las decisiones que he tomado en la vida y por creer siempre en mí. Estos son mis resultados, se los obsequio hoy. Te amo con todo mi corazón.*

*A mi precioso hermanito que la vida me regaló: **Reinier Viamontes Martínez** por ser mi esperanza de seguir adelante. Por ser mi inspiración y por ser esa personita que día a día añoras.  
Sencillamente por ser la base de mi vida. **Tondy** te adoro.*

*A mi novio: **Adrian Gómez García** por ser mi pareja por dos años. Por su paciencia y apoyo incondicional que me ayudo a continuar y culminar mi meta. Por soportarme todas mis malcriadeces. Por ser parte de mí y haber apreciado y cambiado mi vida. Siempre ha estado animándome cuando las cosas no me han ido del todo bien. Te amo.*

*A mis tías preciosas **Iris y Graciela (Cheli)**, por preocuparse siempre, por ayudarme en todo lo que han podido y hasta en las cosas que no han podido también. Las adoro.*

*A mis primos **Margarita, Israel (Bolo) y Yiroannis (Yiro)** por su amor, por quererme tanto. Los quiero mucho.*

*A mis abuelos: **Margarita, Winquer (Papi Chichi)** por estar siempre pendientes de mí y por hacerme tan feliz con su presencia.*

*A **Lorenzo** por su preocupación por mí y por mis estudios, por sus consejos, por estar siempre a mi lado y su apoyo incondicional como si fuera mi propio Padre*

*A mis suegros: **Teresa y Rubén** por su preocupación e incentivarme a cumplir con mis deberes y obligaciones.*

*A mis buenos amigos de Nicaragua: porque a pesar de estar lejos me han apoyado tanto en lo personal como en los estudios, por hacerme reír tanto cada vez que nos encontramos y por saber que los tengo en mi vida. Gracias por ser mis amigos y recuerden que siempre los llevaré presente*

*A mis tres queridas amigas de toda la vida: **Valentina (Valen), Kenia y Deigly** por ser mis favoritas, los hombros a los que siempre me recuesto. Las adoro.*

*A mis amigos la UCI: **Roxana, Mairienis, Marita, Eliza, Yanetsy, Leyni, Zuzi, Yavseny, Dayannis, Anabel, María Elena, Isandra, Ramón, Aramís, Yuri Antonio, Saimel** por estar cuando los he necesitado, en los buenos momentos y en los difíciles también., por tanto apoyo y preocupación.*

*A **Dani y Juli** por su confianza en mí, por su preocupación, por su acogida y hacer con su ayuda más fácil todo este tiempo que he estado tan lejos de mi casa*

*En fin, a todas las personas que quiero y que de una u otra forma hicieron que mi etapa universitaria fuera enriquecedora tanto emocional como intelectualmente. Sin intención de haber dejado de mencionar a alguien.*

***Yiliexy***

*Agradezco a la **Revolución** y a **Fidel** el permitirme estudiar en una escuela como esta.*

*A mis amigos que tantas malacrianzas me aguantaron siempre y aun así me apoyaron incondicionalmente.*

*A toda las personas que siempre confiaron en mí y a las cuales espero no haber decepcionado.*

*A todos los que de una forma u otra me han permitido convertirme en la persona que soy.*

***Hugo***

## DEDICATORIA

*A todos los que me brindaron su apoyo incondicional,*

*A los que siempre estuvieron junto a mí sin importar la distancia,*

*A todos los que me ayudaron,*

*A los que me alentaron,*

*A los que creyeron en mí,*

*Sin ellos hubiera sido realmente difícil llegar hasta aquí.*

*[A mi madre][A mi hermano][A mis abuelos][Mi novio] [A mis amigos]*

*Yily*

*A mi papá que me alentó a estudiar para graduarme.*

*A mi mamá que siempre ha estado ahí cuando la he necesitado y ha sido mi apoyo en todo momento.*

*A mi tía que ha sido mi segunda madre.*

*A toda mi familia.*

*A mi gran amigo Cesar Pérez Castillo sin el cual no hubiera logrado tanto.*

*Hugo*

## RESUMEN

Dado que actualmente en la UCI se está desarrollando un sistema que informatice todos los procesos del Sistema Integral para la Gestión de Recursos Humanos y no existe una interfaz de usuarios que facilite dicha gestión, bajo esta necesidad se decidió, desarrollar la composición de una interfaz web para gestionar mejor la información cumpliendo con los estándares web internacionales, así como con los requerimientos del usuario; mostrando el resultado del trabajo del programador de interfaz, acorde con la planificación para el cumplimiento de la implementación y prueba de cada interfaz de usuario de este sistema.

Una interfaz de usuario no es más que una mediación, entre hombre y máquina, que facilita la comunicación, la interacción, accesibilidad, navegabilidad y eficiencia, entre el ser humano y la computadora. El diseño de interfaces de usuario es un aspecto que ha adquirido relevancia en el desarrollo de sistemas y la calidad del mismo puede influir de manera positiva o negativa en el criterio de los stakeholder sobre el producto, ya que es este aspecto una de los mas representativo a la vista de los usuarios finales.

El Sistema Integral para la Gestión de Recursos Humanos consta con dos módulos **“Selección al Ingreso”** y **“Estructura y Composición”** donde las interfaces de usuario desarrolladas deben ser:

Funcionales, amigables y rápidas.

- ✓ Que cumplan con los requerimientos del sistema así como con los requerimientos del usuario.
- ✓ Que cumpla con los estándares de calidad y diseño internacionales.
- ✓ Que utilicen las tecnologías más novedosas y que hayan sido aprobadas a su vez por el proyecto de la Aduana.

Por tal motivo se pretende con este Trabajo de Diploma una implementación eficiente de estas interfaces de usuario, que va a contribuir en gran medida, al éxito del software que garantizará la gestión de información con el inspector de aduana que trabaje en los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

## PALABRAS CLAVE

- **Interfaz:** Conexión física y funcional entre dos aparatos o sistemas independientes.
- **Capa:** Conjunto de subsistemas que comparten la misma generalidad y de volatilidad de interfaces. Este término hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: **Presentación/** Lógica de Negocio/Datos.
- **Implementación:** Acción de implementar. Este término se refiere a la forma en que se van a codificar las interfaces.

# TABLA DE CONTENIDOS

<b>AGRADECIMIENTOS</b>	II
<b>DEDICATORIA</b>	IV
<b>RESUMEN</b>	V
<b>TABLA DE CONTENIDOS</b>	VII
<b>INTRODUCCION</b>	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b>	6
1.1 INTRODUCCIÓN AL CAPÍTULO	6
1.2 ¿QUÉ ES UNA INTERFAZ GRÁFICA DE USUARIO (GUI, GRAPHICAL USER INTERFACE)?	6
1.2.1 Surgimiento y desarrollo de las GUIs	7
1.2.2 Características de las GUIs	8
1.2.3 Desarrolladores de las GUIs	9
1.3 PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN	10
1.3.1 Algoritmos y Estructuras de datos	11
1.4 PARADIGMAS Y LENGUAJES DE PROGRAMACIÓN	13
1.5 HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS	17
1.5.1 Ext JS	18
1.5.2 Json	19
1.5.3 Ajax	20
1.5.4 DOM	21
1.5.5 CSS	21
1.5.6 Lenguajes de Programación	22
1.5.7 IDEs de Programación	24
1.5.8 Frameworks	26
1.6 PRUEBAS	31
1.7 CONCLUSIONES DEL CAPÍTULO	31
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA</b>	33
2.1 INTRODUCCIÓN AL CAPÍTULO	33

2.2 SOLUCIÓN PROPUESTA-----	33
2.3 CAPA DE PRESENTACIÓN -----	34
2.3.1 <i>Breve descripción de las Interfaces que contiene la capa de Presentación</i> -----	36
2.3.1.1 Módulo “Selección al Ingreso” -----	36
2.3.1.2 Módulo “Estructura y Composición” -----	57
2.4 INTEGRACIÓN CON AJAX -----	69
2.4.1 <i>Helpers básicos de Javascript</i> -----	70
2.4.1.1 Javascript en las plantillas -----	71
2.4.1.2 Actualizando un elemento DOM -----	72
2.4.2 <i>Helpers de Ajax</i> -----	73
2.4.3 <i>Enlaces Ajax</i> -----	75
2.4.4 <i>Formularios Ajax</i> -----	76
2.4.5 <i>Creando efectos visuales</i> -----	77
2.5 ESTRATEGIA PARA LA CAPTURA DE ERRORES -----	77
2.6 CONCLUSIONES DEL CAPÍTULO-----	78
<b>CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA</b> -----	79
3.1 INTRODUCCIÓN AL CAPÍTULO-----	79
3.2 DESCRIPCIÓN DE LAS PRUEBAS DE VALIDACIÓN DE INTERFACES DE USUARIO-----	79
3.2.1 <i>Modulo “Selección al Ingreso”</i> -----	79
3.2.2 <i>Modulo “Estructura y Composición”</i> -----	91
3.3 CONCLUSIONES DEL CAPÍTULO-----	99
<b>CONCLUSIONES GENERALES</b> -----	100
<b>RECOMENDACIONES</b> -----	101
<b>REFERENCIAS BIBLIOGRAFICAS</b> -----	102
<b>GLOSARIO DE TERMINOS</b> -----	104

## INTRODUCCION

El mundo, hoy en día, se ve inmerso en un gran desarrollo científico-técnico, donde las Tecnologías de la Información y las Comunicaciones (TICs) desempeñan un papel trascendental. Estas tecnologías se han convertido en una herramienta al servicio de las estrategias de las aduanas, posibilitando un mayor carácter competitivo, así como la producción de una nueva gama de sistemas que van a satisfacer la gestión, desarrollo sostenible y calidad en cada uno de sus procesos.

En la producción de los distintos sistemas informáticos para la gestión de los procesos aduanales con calidad y acorde con las necesidades de la entidad, hay que destacar que el encargado de gestionar la información en las aduanas es uno de los más importantes, porque permite, entre otras funcionalidades, el control de toda la información fehaciente que favorecerá la toma de decisiones por parte de los directivos y todo ello va a contribuir con el mejor funcionamiento de la entidad, en función del desarrollo de la Aduana y del país.

Para el desarrollo de un sistema es contar con una interfaz válida para todos los usuarios y acorde al contexto de cada proceso identificado, cumpliendo con los requerimientos no funcionales y teniendo en cuenta la apreciación del cliente. Una interfaz inteligente, y bien definida es diseñada específicamente y especialmente de acorde a los principios de cada usuario final. Pueden existir interfaces con propósitos comunes, apariencias diferentes y resultados exitosos para el cliente. Diseñada específicamente para la gente que la usará pues la mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

Cada vez se hace más imperativo la mejor interacción hombre-computadora a través de una adecuada interfaz (Interfaz de Usuario), que brinde tanto comodidad, como eficiencia y es exactamente lo que necesita el Sistema Integral para la Gestión de Recursos Humanos, lo que representa un gran problema pues cuando alguien usa una herramienta, o accede e interactúa con un sistema informático, suele haber "algo" entre ese alguien y el objeto de interacción; ese algo, que es a la vez un límite y un espacio común entre ambas partes, es la interfaz .

Interactuamos con el mundo que nos rodea a través de cientos de ellas. Muchas son tan conocidas y aceptadas, que ni siquiera nos damos cuenta de que existen. Sin embargo muchas de ellas, por nuevas, desconocidas o mal diseñadas, son visibles. ¿Cuántas veces no encuentran lo que buscan o no saben cómo hacer lo que quieren? Esta situación es el resultado de una mala interfaz, que a su

vez genera un problema de usabilidad. El mejor sistema o la herramienta perfecta, es inútil si no podemos interactuar. Si el sistema que se está desarrollando está generando un nuevo medio de comunicación se necesita una interfaz con calidad pues es la que juega el papel fundamental.

En nuestro país se desarrollan una serie de proyectos, regidos en su mayoría, por la Universidad de las Ciencias Informáticas (UCI) y entre los más relevantes se encuentra un nuevo Sistema Integral para la Gestión de Recursos Humanos, que pueda ser utilizado por cualquier aduana del país, con el fin de erradicar los problemas que presentan en cuanto: la gestión de la información, que en su mayoría son de organización y control precisamente por carecer este sistema de una Interfaz de Usuario que agilice todos estos procesos.

En la construcción de los sistemas informáticos, en la actualidad, se aboga por la utilización de alguna de las metodologías de desarrollo de software, que sirven para dirigir, organizar y controlar el proceso de desarrollo de estos sistemas y brindan mayores posibilidades de cumplimentar el trabajo de forma eficiente y con calidad. Entre los roles se encuentra el rol de arquitecto, que es aquella persona encargada de definir la arquitectura que se va a utilizar para desarrollar el sistema y en este caso, se emplea un estilo arquitectónico que va a dar solidez a la misma, como eje central del sistema: el Modelo Vista Controlador.

En el modelo vista controlador, la vista es el medio donde se manipula toda la información de forma visual, el controlador es el encargado de interpretar todas las acciones sobre el sistema e informa tanto a la vista como al modelo para que actúen en consecuencia y el modelo es el encargado de administrar tanto los datos como el comportamiento de la aplicación.

Para el desarrollo eficiente de cada una de estas capas, el programador debe utilizar los artefactos elaborados por el arquitecto, los analistas y los diseñadores, estos dos últimos, son roles que también definen la metodología. Entre los principales artefactos que necesita el programador de la capa de Presentación, se encuentran: el Modelo de Datos, el Modelo de Diseño, el Prototipo de Interfaz.

Teniendo en cuenta todo lo expuesto, se puede definir el **Problema a resolver:**

¿Cómo implementar una interfaz amigable, funcional y rápida, para los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos?

**Objeto de estudio:** Interfaz de Usuario del Sistema Integral para la Gestión de Recursos Humanos.

Para lograr todo ello, se plantea como **objetivo general de la investigación**: Implementar una interfaz funcional, amigable y rápida, en la capa de Presentación de los módulos de “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

**Campo de acción**: Interfaz de usuario de los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

**Hipótesis**: Si se cumplen con los requerimientos del sistema así como con los requerimientos del usuario y además con los estándares de calidad y diseño internacionales entonces se obtendrá una interfaz funcional, amigable, rápida y agradable para el usuario que favorecerá el incremento de la eficiencia y rapidez en los procesos de Gestión.

Las **tareas** para desarrollar la investigación, son las siguientes:

- ✓ Valorar y analizar el estado del arte del rol de programador de interfaz fundamentalmente en el desarrollo de la interfaz de usuario de los principales sistemas de gestión.
- ✓ Utilizar el Proceso Personal del Software: Personal Software Process (PSP) para lograr un aprovechamiento máximo del tiempo que se dispone para implementar las interfaces.
- ✓ Desarrollar métodos en las clases de la capa de presentación, que favorezcan las validaciones y el trabajo con los distintos tipos de variables, brindando mayor eficiencia y uniformidad en la implementación de las interfaces.
- ✓ Analizar los requisitos que deben cumplir las interfaces acorde con la descripción de los casos de usos y el prototipo de interfaz.
- ✓ Implementar la interfaz de usuario de los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.
- ✓ Desarrollar las pruebas, con los casos de uso de prueba correspondiente, que posibilitarán la validación de las interfaces.

Para dar cumplimiento a las distintas tareas antes mencionadas, se pusieron en práctica, los siguientes **métodos de investigación**:

### Métodos teóricos:

- ✓ *Analítico – sintético*: este método posibilita el procesamiento de toda la información y poder sintetizar y diferenciar cada una de ella, enfocada hacia la investigación, además de poder determinar las características que debe tener el desarrollo de la Interfaz .
  
- ✓ *Inductivo – deductivo*: permite elaborar los estándares, tanto de codificación y diseño gráfico, como las clases a utilizar por las interfaces y a su vez, determinar las especificaciones de cada una de ellas.
  
- ✓ *Histórico – lógico*: este método posibilita conocer los antecedentes y tendencias actuales del las interfaces gráficas y del desempeño de un programador de la capa de Presentación, principalmente en los softwares de gestión.

### Métodos empíricos:

- ✓ *Medición*: este método posibilita establecer comparaciones en el desarrollo del PSP del programador de la capa de Presentación.
  
- ✓ *Experimento*: este método favorece el desarrollo de las pruebas para la validación de las interfaces.

### Aportes prácticos que se esperan del trabajo:

El trabajo desarrollado debe dar la posibilidad de contar con:

- ✓ Una integración de la Interfaz con otras unidades funcionales desarrolladas para el Sistema Integral para la Gestión de Recursos Humanos.
  
- ✓ La implementación y certificación de una interfaz funcional, amigable y rápida, los módulos de "Selección al Ingreso" y "Estructura y Composición" del Sistema Integral para la Gestión de Recursos Humanos.

**Estructura de la Tesis:**

El Trabajo de Diploma está estructurado en tres capítulos.

En el **Capítulo 1** Se aborda una propuesta de diseño de interfaz para los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos. Se abordó sobre el surgimiento y desarrollo de la interfaz gráfica en los principales software de gestión de información. Se presentó el resultado de un estudio realizado sobre las principales técnicas y tecnologías de programación web que se utilizan tanto en la UCI, como en el país y a nivel internacional.

Además se hace una breve valoración, de las mismas para dar cumplimiento a las tareas del rol de programador de interfaz enfocadas fundamentalmente en los lineamientos que rigen su desarrollo dentro del proyecto productivo. Se explica la estrategia de investigación y se hace una breve crítica y valoración, de las técnicas, lenguajes, metodologías, plataforma y procedimientos utilizados para dar cumplimiento a las tareas del programador de interfaz.

En el **Capítulo 2** En este capítulo se dio una panorámica de cómo se da respuesta al problema, la solución propuesta; para ello se abordará temas, sin los cuales no hubiese sido posible la realización del sistema. Como detalle principal se habló de la capa de Presentación la cual brinda interfaces que le permiten al usuario intercambiar con el sistema y la utilización de los componentes que nos brinda la librería Extjs.

Se dio una breve descripción de las interfaces de usuario que contiene la capa de Presentación, cumpliendo los requisitos de forma correcta de ambos módulos.

También trató sobre las estrategias de integración con AJAX que satisfacen nuestro sistema y la importancia de las mismas para mayor organización y rapidez de la aplicación, así como la estrategia para la captura de errores en los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

En el **Capítulo 3** se exponen los resultados de las pruebas de validación de interfaces realizadas a cada uno de los CU implementados en la capa de Presentación de los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

Además de poseer Conclusiones Generales, Recomendaciones, Referencias Bibliográficas y Anexos.

# 1

## Capítulo 1: Fundamentación Teórica

### 1.1 Introducción al capítulo

En este capítulo se realiza un estudio acerca del surgimiento y desarrollo de la interfaz gráfica, fundamentalmente en los principales software de gestión de información, así como de las principales técnicas o estilos de programación que se utilizan tanto en la UCI, como en el país y a nivel internacional.

Además se exponen un conjunto de conceptos de las terminologías, que representan las principales tareas que debe realizar un programador de interfaz dentro de un proyecto productivo. Se realiza una breve crítica y valoración, de las técnicas, plataforma, librerías y procedimientos utilizados para dar cumplimiento a las tareas del programador de interfaz.

### 1.2 ¿Qué es una Interfaz Gráfica de Usuario (GUI, Graphical User Interface)?

El término *interfaz*, proviene del inglés *interface* y pudiera definirse, de forma general, como el medio que posibilita la interacción entre dos elementos para que puedan trabajar juntos. Sin embargo, en la esfera de la informática, hay que destacar que existen diversos tipos de interfaces que trabajan a distintos niveles. Existen las interfaces que hacen posible el trabajo del software con el hardware, es decir, que posibilita a los sistemas operativos el funcionamiento con los dispositivos de hardware, tanto interno como externos.

También existe la interfaz de línea de comandos para interpretar las sentencias introducidas por el usuario, la interfaz controlada por menús, que le brinda al usuario mayores funcionalidades para la interpretación de combinaciones de teclas en el trabajo con la aplicación y la interfaz gráfica de usuario o GUI, que además de controlar las opciones de los menús, también posibilita controlar la posición, el tamaño y el contenido de las áreas de trabajo, en pantalla.

La GUI es definida por Lewis y Rieman de la siguiente forma:

*“Las interfaces gráficas de usuario son aquellas que incluyen cosas como menús, ventanas, teclado, ratón y algunos otros sonidos que la computadora hace, en general, todos aquellos canales por los cuales se permite la comunicación entre el hombre y la computadora.” [Lewis-Rieman, 1993]*

También existen otras definiciones tales como:

*“...tipo de entorno que permite al usuario elegir comandos, iniciar programas, ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú...” [Stephanidis, 2001]*

*“...es aquella parte de un programa que comunica al usuario con el programa mediante representaciones gráficas...” [Stephanidis, 2001]*

La definición más acertada para esta terminología, según la opinión del autor, acorde con la esfera informática y el desarrollo de software, es la siguiente:

“La Interfaz Gráfica de Usuario es un tipo de interfaz que mediante el uso del lenguaje visual, sustentado por un conjunto de imágenes y componentes gráficos, muestra en la pantalla una información específica así como un conjunto de acciones posibles, posibilitando a los usuarios una interacción con un sistema informático.”

### 1.2.1 Surgimiento y desarrollo de las GUIs

La idea de una interfaz gráfica de usuario para un computador u ordenador, surge incluso, mucho antes de que la tecnología estuviera disponible para ello. El Memex (Memory Expander) era una máquina que Vannevar Bush había ideado, pero nadie pudo materializarlo y en síntesis, posibilitaba la búsqueda y reproducción de archivos microfilmados y daba además la posibilidad a los usuarios de tomar anotaciones en los márgenes. Luego, en 1945 Bush retoma sus ideas que inspiraron a Douglas Engelbart, quien construyó la máquina. [Bush, 1945]

Douglas recibe el apoyo por parte de las Fuerzas Aéreas de los Estados Unidos e inventa el ratón del ordenador y la primera interfaz gráfica en 1960. Culminó su trabajo en 1968 con una demostración ante miles de personas y es considerado como el padre de la Interfaz Gráfica de Usuario. [Reimer, 2005]

Esa interfaz gráfica creada por Douglas en los laboratorios de Xerox, fue introducida en las computadoras de Apple - Macintosh en 1984 y no fue hasta el 1993 y dado por la necesidad de eliminar muchas de las barreras existentes hombre – ordenador, que se llevó a la población, con la primera versión popular de Windows 3.0 como sistema operativo. [Reimer, 2005]

En la actualidad la GUI es un elemento fundamental para la producción de software, porque refleja el rostro del mismo y en cierto grado, delimita su carácter competitivo. Las GUIs de los sistemas de gestión de inventarios, son interfaces muy amigables, rápidas, funcionales y flexibles, porque su misión fundamental es brindar mayores facilidades a los usuarios en el trabajo con el sistema.

Ejemplo de ello se tiene el sistema francés Geode GX, perteneciente a la Sage Adonix, líderes en el mercado de aplicaciones especializadas en gestión logística, con instalaciones en más de 350 plantas en Francia, Europa y Norteamérica; también existen otros sistemas importantes producidos por los italianos, mexicanos y argentinos, como el WMS (Warehouse Management Software) de la System, el ADMAN de la Siman (Sistemas Maestros de Negocios) y el sistema ISIS de la ALIANZA DATAHOUSE-COMPANY respectivamente.

### 1.2.2 Características de las GUIs

En la actualidad, una buena interfaz gráfica de usuario debe contar con un conjunto de características que podrían sintetizarse en:

- ✓ Facilidad de comprensión, aprendizaje y uso.
- ✓ Representación fija y permanente de un determinado contexto de acción.
- ✓ Facilidad de identificación del objeto de interés.
  
- ✓ Diseño funcional, mediante el establecimiento de menús, barras de acciones e iconos de fácil acceso.
- ✓ Las interacciones se basarán en acciones físicas sobre elementos de código visual o auditivo (iconos, botones, imágenes, mensajes de texto o sonoros, barras de desplazamiento y navegación...) y en selecciones de tipo menú con sintaxis y órdenes.
- ✓ Las operaciones serán rápidas, incrementales y reversibles, con efectos inmediatos.
- ✓ Existencia de herramientas de Ayuda y Consulta.

- ✓ Tratamiento del error bien cuidado y adecuado al nivel de usuario.

### 1.2.3 Desarrolladores de las GUIs

Para desarrollar cada una de las capas que definen los estilos arquitectónicos del Sistema Integral para la Gestión de Recursos Humanos, la metodología RUP define grupos de trabajadores: *Analistas*, *Desarrolladores*, *Probadores*, *entre otros*, así como un conjunto de artefactos y actividades a realizar, que precisan las responsabilidades de estos trabajadores. Entre los que clasifican como desarrolladores se encuentran los *Programadores*, que son los encargados de implementar y probar los componentes que necesita el software, acorde con los estándares adoptados por el proyecto, para la integración en subsistemas más grandes. [RUP, 2003]

Es preciso que el programador reúna un conjunto de características que le posibilite personalizar su trabajo, sin aislarse de las guías adoptadas para el mismo. La originalidad y creatividad, son aspectos claves para desempeñar este trabajo, así como también lo son la capacidad de superación con facilidad, la investigación de forma independiente, la capacidad de analizar y observar los sucesos dentro y fuera de su entorno de trabajo y sobre todo, la aplicación de la lógica en todo momento.

Según RUP:

Las actividades fundamentales que realiza el programador son:

- ✓ Implementar los elementos de diseño y prueba.
- ✓ Desarrollar pruebas de implementación y ejecución.
- ✓ Analizar el comportamiento de componentes en tiempo de ejecución.

Los artefactos principales a desarrollar por el programador son:

- ✓ Elementos y subsistemas de implementación.
- ✓ Elementos de prueba.

Para que las interfaces gráficas de usuario reúnan todas las características que le permitan la obtención del éxito al software que representan, deben ser implementadas cuidadosamente. Con frecuencia estas implementaciones se realizan en un tiempo considerablemente corto, pero si no se

tienen en cuenta aspectos como patrones de diseño, reusabilidad, seguridad y optimización, entre otros, las interfaces se tornan poco eficientes. En esta capa los programadores deben hacer gala de todas sus habilidades y conocimientos, porque de ello depende en un por ciento elevado, la aceptación del software.

### 1.3 Principios básicos de la programación

La programación, en términos informáticos, se muestra como la codificación de las órdenes y datos que permiten la creación de un programa o aplicación, pero antes de llevar a cabo la programación de una aplicación o sistema, es importante que el programador realice una serie de actividades que posibiliten su desarrollo de forma eficiente. En todo proceso de construcción o elaboración de un producto, existe un elemento fundamental, donde se van a trazar los lineamientos para lograr el objetivo del proceso: el *diseño del producto*.

En el caso de los programas y aplicaciones, el diseño consta de distintas etapas de desarrollo. La realización de un **análisis** exhaustivo de cada una de las situaciones tanto explícitas como implícitas que contenga el problema a resolver. El **diseño** y creación o utilización de la base de la programación: **los algoritmos**, apoyado en el análisis. Una vez que se tienen las bases del programa, se procede a realizar la **codificación** o **implementación** de los algoritmos en un lenguaje de programación y reflejando las ideas desarrolladas en las etapas de análisis y diseño. Posterior a la codificación, es preciso que el programa sea llevado al lenguaje de la máquina, para su ejecución, es decir, que sea **interpretado** o **compilado** y **ejecutado**. [Figueroa, 2005]

Durante el desarrollo de las etapas anteriores pueden existir errores que serán buscados y eliminados en las etapas de **verificación y depuración**, etapas fundamentales para la culminación de un programa eficiente, así como la **documentación** o etiquetado del programa, para hacerlo más comprensible. [Figueroa, 2005]



Figura 1. 1: Esquema de las etapas para el diseño de un programa.

Existe una octava etapa que está sujeta principalmente a programas o aplicaciones de mayor envergadura, es decir, que presentan mayor grado de complejidad, tanto en la programación como en la utilización o vinculación con otros sistemas. Este tipo de programas o aplicaciones necesitan, en la mayoría de los casos, de un mantenimiento donde puedan realizarse nuevos ajustes y modificaciones, siempre teniendo en cuenta que no deben alterar el funcionamiento de los mismos. [Figuerola, 2005]

### 1.3.1 Algoritmos y Estructuras de datos

El diseño de algoritmo, es la etapa donde se va a resolver el problema en cuestión, aunque no por sí solo, sino con la aplicación correcta de todas las operaciones que son dictadas por el algoritmo. La palabra *algoritmo* se deriva del nombre del gran matemático árabe Mahommed ibn Musa al-Jwarizmi (c. 780-c. 835), debido a sus avances en el trabajo con el álgebra y los algoritmos, con fines matemáticos. Es preciso destacar que los algoritmos no se centran en una ciencia o materia específica, son aplicables a la mayoría de las acciones que realiza el hombre. [Cormen-Leiserson-Rivest-Stein, 2001]

Existen diversas definiciones de algoritmo entre las cuales se encuentran:

*“...es cualquier procedimiento computacional bien definido, que toma algún valor, o juego de valores, como la entrada y produce algún valor, o juego de valores, como el rendimiento. Un algoritmo es así una sucesión de pasos que transforman la entrada en el rendimiento.” [Cormen-Leiserson-Rivest-Stein, 2001]*

*“Un algoritmo es un juego finito de instrucciones que, si se siguen, logra cumplimentar una tarea particular” [Pinson - Wiener, 2000]*

Teniendo en cuenta todos estos criterios sobre los algoritmos, se puede definir a un algoritmo como un método sustentado por un conjunto de instrucciones finitas, que utilizando un grupo de elementos de entradas, produce otro grupo de elementos resultantes para dar solución a un problema determinado, independientemente del lenguaje de programación que se utilice para desarrollarlo.

### **Características de los algoritmos:**

- ✓ Debe tener un punto particular de inicio.
- ✓ No debe ser ambiguo.
- ✓ Debe ser general.
- ✓ Debe ser finito en tamaño y en tiempo de ejecución.
- ✓ Legible y sencillo de modificar-ampliar.
- ✓ Eficiente, definido (“n ejecuciones - igual resultado”) y óptimo. [Horta-Ostos-Gutiérrez, 1991]

Es importante destacar que los algoritmos, como eje central de la metodología de la programación, frecuentemente demandan una compleja organización de sus datos y una de las formas más eficiente y organizada de manipular estos datos es precisamente con la utilización de las estructuras de datos. Las estructuras de datos van a facilitar todo un marco lógico para el trabajo de los datos en función del problema que se vaya a tratar y el algoritmo para resolverlo.

Se considera que la definición más acertada de una estructura de datos y a la cual se adscribe el autor, es la siguiente:

“...una manera de almacenar y organizar los datos de forma ordenada, para facilitar el acceso y las modificaciones a los mismos. Es importante tener conocimiento de las ventajas y desventajas de cada una de las estructuras de datos, para enfocar su uso eficiente en un problema, acorde con sus características, porque no todas las estructuras de datos funcionan bien con todos los tipos de problemas.” [Cormen-Leiserson-Rivest-Stein, 2001]

Entre las estructuras de datos de uso más frecuente, se encuentran los Arreglos, Listas Enlazadas, Pilas, Colas, Árboles y Grafos. Cada una de ellas con sus respectivas especificaciones y variantes.

Las estructuras de datos y los algoritmos están indisolublemente unidos. Es importante identificar y comprender correctamente los algoritmos que se vayan a utilizar en la solución a un problema determinado, para poder seleccionar y ajustar en caso necesario, estructuras de datos que organicen apropiadamente el trabajo.

### 1.4 Paradigmas y Lenguajes de programación

La palabra *paradigma* proviene del griego *paradigma*, que significaba originalmente: *ejemplo ilustrativo* y fue el filósofo Thomas Samuel Kuhn (1922-1996), quien extendió la definición de la palabra:

*"Considero a los paradigmas como realizaciones científicas universalmente reconocidas que, durante cierto tiempo, proporcionan modelos de problemas y soluciones a una comunidad científica".* [Fuente, 2000]

Los paradigmas de programación según Bobrow y Stefik son: *una forma de organizar programas sobre las bases de algún modelo conceptual de programación y un lenguaje apropiado para que resulten claros los programas escritos en ese estilo.* [Bobrow-Stefik, 1986]

Según R. Floyd, existen tres tipos de paradigmas:

- ✓ *Los que soportan técnicas de programación de bajo nivel.*
- ✓ *Los que soportan métodos de diseño de algoritmos.*
- ✓ *Los que soportan soluciones de programación de alto nivel.*

Teniendo en cuenta que un paradigma va a definir un conjunto de reglas, patrones y estilos de programación que son usados por los lenguajes de programación que usan ese paradigma y acorde con los tipos de paradigmas definidos por Floyd, se pueden destacar cuatro paradigmas fundamentales:

**1. Paradigma funcional:** en este paradigma el proceso computacional se lleva a cabo a través de la evaluación de expresiones.

**Características:**

- ✓ Definición de funciones.
- ✓ Funciones como datos primitivos.
- ✓ Valores sin efectos laterales, no existe la asignación.
- ✓ Programación declarativa.

**2. Paradigma lógico:** en este paradigma la programación trata de representar el conocimiento mediante relaciones (predicados) entre objetos (datos).

**Características:**

- ✓ Definición de reglas.
- ✓ Unificación como elemento de computación.
- ✓ Programación declarativa.

**3. Paradigma imperativo o procedural:** los lenguajes que cumplen con este paradigma presentan un estado implícito que es modificado mediante instrucciones o comandos del lenguaje.

**Características:**

- ✓ Definición de procedimientos.
- ✓ Definición de tipos de datos.
- ✓ Chequeo de tipos en tiempo de compilación.
- ✓ Cambio de estado de variables.
- ✓ Pasos de ejecución de un proceso

**4. Paradigma orientado a objetos:** este paradigma es uno de los más empleados en la actualidad. Define los programas en términos de clases y objetos.

**Características:**

- ✓ Definición de clases y herencia.

- ✓ *Objetos como abstracción de datos y procedimientos.*
- ✓ *Polimorfismo y chequeo de tipos en tiempo de ejecución.* [Floyd, 1979]

Es preciso destacar que no hay un paradigma de programación que sea el mejor para todo tipo de aplicaciones. Por ejemplo, la programación orientada a reglas sería la mejor para el diseño de una base de conocimiento, y la programación orientada a procedimientos sería la más indicada para el diseño de operaciones de cálculo intensivo. Actualmente uno de los paradigmas más utilizados tanto a nivel internacional como en nuestro país, es el orientado a objetos, porque teniendo en cuenta las abstracciones que implementa este paradigma de programación, se puede afirmar que se adecua, en mayor grado, a una gran diversidad de aplicaciones. [Booch, 1996]

Los paradigmas de programación que se emplean en el desarrollo del SUA son:

Paradigmas *Orientado a Objetos* y *Orientado a Aspectos*, este último, como una alternativa a la solución de algunos de esos problemas que provocarían una disminución de la calidad final del software, como lo son la abstracción y el encapsulamiento de conceptos, que no forman parte de la funcionalidad básica de los sistemas, tales como debugging, sincronización, distribución, seguridad, administración de memoria, y otros.

Los lenguajes y los paradigmas de programación se encuentran estrechamente relacionados. De la misma forma en que existen paradigmas distintos, con ideas comunes, existen lenguajes que soportan más de un paradigma.

Según la definición de la Encyclopedia of Computer Science (Encyclopedia of Computer Science, 4th Edition, Anthony Ralston (Editor), Edwin D. Reilly (Editor), David Hemmendinger (Editor), Wiley, 2007. Disponible en la biblioteca politécnica con identificador: POE R0/E/I/ENC/RAL):

*"A programming language is a set of characters, rules for combining them, and rules specifying their effects when executed by a computer, which have the following four characteristics:*

- ✓ *It requires no knowledge of machine code on the part of the user*
- ✓ *It has machine independence*
- ✓ *Is translated into machine language*
- ✓ *Employs a notation that is closer to that of the specific problem being solved than is machine code. "*

Es decir:

*“Un lenguaje de programación es un conjunto de caracteres, reglas para combinarlos, y reglas que especifican sus efectos cuando son ejecutados por una computadora, que tienen las cuatro características siguientes:*

- ✓ *No requieren del conocimiento del código de la máquina para la parte de los usuarios.*
- ✓ *Tiene independencia de la máquina.*
- ✓ *Se traduce en el idioma de la máquina.*
- ✓ *Empleos de una notación que se acerca al problema específico a resolverse, que es el código de la máquina.”*

También es preciso destacar ideas como las de Abelson y Sussman acerca de los lenguajes de programación:

*“A powerful programming language is more than just a means for instructing a computer to perform tasks. The language also serves as a framework within which we organize our ideas about processes. Thus, when we describe a language, we should pay particular attention to the means that the language provides for combining simple ideas to form more complex ideas.”*

*“Un lenguaje de programación poderoso es más que un simple medio para instruir a una computadora en la realización de las tareas. El lenguaje también sirve como un medio donde podemos organizar nuestras ideas sobre los procesos. Así, cuando nosotros describimos un lenguaje, debemos prestar particular atención a los recursos que brinda el lenguaje, para combinar ideas simples con ideas más complejas.” [Abelson - Sussman, 1996]*

Según la definición de los doctores en ciencias Montse Ibañez y Germán Gutiérrez y a la cual se adscribe el autor, un lenguaje de programación es: un tipo de lenguaje que permite comunicar conceptos entre dos elementos fundamentales (máquina-humanos) y como cualquier lenguaje consta de:

- **Alfabeto:** caracteres alfanuméricos (alfabéticos + numéricos) y otros especiales como =, +, -, \*, , (, ), ., :, {, etc.

- **Léxico:** vocabulario:
  - Palabras predefinidas (reservadas)
  - Resto palabras: reglas léxicas (identificadores válidos)
  
- **Sintaxis:** reglas gramaticales que determinan de manera estricta como formar las instrucciones (sentencias).
  - Errores sintácticos: el programa no compila y/o no se puede interpretar
  
- **Semántica:** Significado tanto del léxico como de las instrucciones. El programa, al ejecutarlo, produce el resultado deseado. [Ibañes – Gutiérrez, 1998]

Entre los años 1954 y 2000, se documentaron alrededor de 2500 lenguajes de programación, pero de ellos hubo una decena que influyeron en el desarrollo de los lenguajes posteriores y que se elaboraron entre 1954 y 1972, es decir, en la primera época de los lenguajes de programación.

Los lenguajes de programación se dividen en tres grandes grupos: los lenguajes de máquina, que son aquellos escritos en código binario, los de bajo nivel o ensamblador, que son escritos en nemotécnicos y los de alto nivel que han sido diseñados para lograr una mejor comprensión en la escritura y lectura de programas. [Bonanata, 2003]

Entre los lenguajes de alto nivel, que han posibilitado la creación de aplicaciones, con mayores prestaciones y facilidades en su desarrollo, apoyándose en una característica fundamental que es la independencia entre ellos y la arquitectura de la máquina u ordenador; se puede destacar al Visual Basic, C, C++, Delphi, CSharp y Java, destacándose en mayor medida C, CSharp y Java. En la actualidad, nuestra universidad consta de un conjunto de proyectos productivos donde un gran número de aplicaciones, son desarrolladas con la utilización de estos lenguajes.

### 1.5 Herramientas y Tecnologías utilizadas

Con el avance de las tecnologías y de internet cada vez hay más instituciones que orientan sus aplicaciones hacia la tecnología web. El cliente hace una petición al servidor para que le envíe una página web, este analiza la extensión de la página solicitada y la devuelve al cliente una vez que la

localiza en su sistema de archivos y ejecuta el intérprete de dicha extensión. La variedad de lenguajes y formatos en los que se puede encontrar codificados las páginas es bien variada, así se encuentran las extensiones como HTML, PHP, HML, XML, etc.

### 1.5.1 Ext JS

**Ext JS**, se comenzó a desarrollar a principios del 2006 por **Jack Slocum** como un conjunto de extensiones para la librería Yahoo! User Interface (YUI), estas extensiones recibieron el nombre de **yui-ext**. El otoño del 2006 la librería gana tanta popularidad que cambia su nombre a **Ext** (con licencia BSD) como un reflejo de su madurez e independencia.

Ext. JS se usa para interfaces avanzadas, contiene casi todo lo que se pueda imaginar y nace como solución a tareas comunes -pero complejas. Está completamente Orientado a Objetos. Cada objeto contiene lo típico: propiedades, métodos, eventos...etc.

**Ext** basa toda su funcionalidad en **JS** a través de librerías ya muy conocidas: YUI, jQuery y Prototype/Script.aculo.us y un *core* interno poderoso. Así, en tiempo de ejecución carga y crea todos los objetos HTML a través del uso intenso de DOM.

Ventanas, mensajes emergentes, grids, *date pickers* y un sin numero de utilidades son todas creadas en tiempo de ejecución. Los datos son obtenidos con mucho AJAX a través de XML y/o JSON.

#### Ventajas:

- ✓ La orientación a objetos intensa te hará modular todos tus scripts (por si es que a estas alturas ya no lo hacías).
- ✓ El diseño está completamente separado de la funcionalidad.
- ✓ Funciones comunes como validación, comboboxes editables, ventanas arrastables (con minimizar y maximizar), grillas editables, son muy fáciles de implementar.
- ✓ Buena y amplia documentación, así como también su comunidad.

#### Desventajas:

- ✓ Crear un sistema serio con esta herramienta requiere un previo uso prolongado, ya que te perderás con muchos nuevos objetos en su extensa y bien documentada API (que por cierto

también hace gala de sus mismas librerías). El tiempo de aprendizaje puede llegar a compararse con a aprender a programar en un lenguaje nuevo.

- ✓ Al estar todo tu sitio en **JS**, no podrá ser accesible para los buscadores, limitando su uso a sistemas y no sitios web.
- ✓ Si existiese algún objeto que desearas y no existiera, te verás en la compleja tarea de crear un nuevo objeto (sólo apto para programadores **JS** avanzados).

### Entre las bondades de este framework se tiene:

- ✓ Modelo de Componentes.
- ✓ Modelo de Contenedores.
- ✓ Capas.
- ✓ Grid.
- ✓ PlantillasX.
- ✓ Vistas de Datos.
- ✓ Sirve de puente entre las librerías JS más usadas (Prototype, JQuery, YUI). Debido a que se inicio como una extensión de YUI esta presenta una cierta ventaja de compatibilidad respecto a las otras dos.

### Soporta los navegadores:

- ✓ Internet Explorer 6+
- ✓ FireFox 1.5+ (PC, Mac)
- ✓ Safari 2+
- ✓ Opera 9+ (PC, Mac)

### 1.5.2 Json

JSON (JavaScript Object Notation) es un formato sencillo para intercambiar datos. Consiste básicamente en un array asociativo de JavaScript que se utilizar para incluir información del objeto. JSON ofrece 2 grandes ventajas para las interacciones Ajax: es muy fácil de leer en JavaScript y puede reducir el tamaño en bytes de la respuesta del servidor.

El formato JSON es el más adecuado para la respuesta del servidor cuando la acción Ajax debe devolver una estructura de datos a la página que realizó la llamada de forma que se pueda procesar con JavaScript. Este mecanismo es útil por ejemplo cuando una sola petición Ajax debe actualizar varios elementos en la página.

Para actualizar los varios elementos, la respuesta Ajax podría consistir en una única cabecera JSON, mediante algunas pocas instrucciones de JavaScript se puede interpretar la respuesta del servidor y actualizar varios elementos de la página de forma seguida.

El protocolo HTTP permite que la respuesta JSON se pueda enviar como una cabecera de la respuesta. Como la respuesta no tiene ningún contenido, la acción envía solo la cabecera de forma inmediata. De esta forma, se evita completamente la capa de la vista y es muy rápido pero además con una respuesta más pequeña.

JSON se ha convertido en un estándar en el desarrollo de aplicaciones web. Los servicios web proponen la utilización de JSON en vez de XML para permitir la integración de servicios en el navegador del usuario en vez de en el servidor. El formato JSON es seguramente la mejor opción para el intercambio de información entre el servidor y las funciones JavaScript.

### 1.5.3 Ajax

AJAX, acrónimo de JavaScript asíncrono y XML, es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

**AJAX es una combinación de tres tecnologías ya existentes:**

- ✓ XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- ✓ Modelo de Objetos de Documento (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.

- ✓ El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- ✓ XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano y JSON

Como el DHTML, LAMP o SPA, AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

### 1.5.4 DOM

El Modelo de Objetos de Documento, frecuentemente abreviado **DOM**, es una forma de representar los elementos de un documento estructurado (tal como una página web HTML o un documento XML) como objetos que tienen sus propios métodos y propiedades.

En efecto, el DOM es una API para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes como ECMAScript (Javascript).

### 1.5.5 CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “*documentos semánticos*”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Mientras que el lenguaje HTML/XHTML se utiliza para *marcar* los contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc.

### Las ventajas de utilizar CSS son:

- ✓ Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ✓ Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- ✓ Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- ✓ El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

### 1.5.6 Lenguajes de Programación

**Java Script:** Es un lenguaje de script basado en objetos, que se apoya en el modelo de prototipos. Es muy popular por su uso en sitios Web. Muy utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web lo constituye JavaScript. Con este se pueden crear efectos visuales para mostrar contenido de forma dinámica. También nos permite ejecutar instrucciones como respuesta a las acciones del usuario con lo que se puede crear páginas interactivas. Nos permite la programación de pequeños scripts pero también de programas más grandes con elementos de orientación a objetos, con funciones, estructuras de datos complejas, etc.

Además pone a disposición del programador todos los elementos que forman la página web para que este pueda accederá ellos y modificarlos dinámicamente. Cuando se usa en Web, se conecta a través de su interfaz DOM (Document Object Model) a las aplicaciones tanto del lado del cliente como del servidor. Entre las principales tecnologías para interactuar con DOM que usan JavaScript esta AJAX y DHTML.

**PHP (Hypertext Preprocessor):** Las características más destacadas de PHP son su facilidad de aprendizaje y que es software libre, su amplio soporte a manejadores de bases de datos lo han convertido en uno de los lenguajes más populares para programación Web. Comparte características

con otros lenguajes de programación como C y Perl y recientemente con Java. Además de ser usado en web, puede usarse desde *la línea de comandos* y *para* construir GUIs con PHP-GTK.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

### **Características:**

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas:

- ✓ Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- ✓ Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- ✓ Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- ✓ Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- ✓ Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- ✓ El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- ✓ Con PHP se puede hacer cualquier cosa que se pueda realizar con un script, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con páginas webs dinámicas es aquel que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para propósitos particulares.

También nos ofrece una gran variedad de funciones que nos permiten desarrollar múltiples funcionalidades que van desde enviar un e-mail, subir un archivo (upload), crear una imagen en tiempo de ejecución, interactuar con diversos protocolos de comunicación, interactuar con documentos XML, autenticación, creación dinámica de documentos PDF, entre muchas otras cosas.

Las principales características de PHP son: su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

### Seguridad en PHP

PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto.

PHP ha sido diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI, Perl o C y con la correcta selección de las opciones de configuración de tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita. Ya que existen diferentes modos de utilizar PHP, existe también una multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes aplicaciones, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

### 1.5.7 IDEs de Programación

➤ **Zend Studio for Eclipse** se convierte así en el IDE (Entorno de Desarrollo Integrado) para PHP más potente del mercado, ofreciendo así al desarrollador profesional de PHP la potencia de Zend Studio y el soporte multilenguaje de Eclipse y su enorme conjunto de extensiones (plugins).

**En concreto, Zend Studio para Eclipse aporta:**

Las ventajas de Eclipse/PDT:

- ✓ Una comunidad de millones de usuarios y miles de desarrolladores cientos de plugins
- ✓ Soporte multi-lenguaje en una única herramienta (Eclipse)
- ✓ Coloreado de sintaxis PHP, autocompletado de código e inspección de métodos y atributos
- ✓ Soporte básico de depuración de scripts PHP

Valor añadido de Zend Studio:

- ✓ Soporte de depuración más avanzada (permitiendo debugging local como remoto)
- ✓ Soporte a la refactorización de código
- ✓ Análisis de rendimiento y Code Coverage
- ✓ Soporte de baterías de test PHPUnit
- ✓ Integración mejorada con Zend Framework
- ✓ Posibilidad de edición HTML WYSIWYG
- ✓ Integración con Zend Platform
- ✓ Herramientas de migración para los usuarios de Zend Studio
- ✓ Integración BIRT
- ✓ Instalación simplificada

Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido".

➤ **Aptana Studio** es un IDE de desarrollo para aplicaciones de la web 2.0, gratuito, código libre, con soporte Ajax, PHP, Ruby on Rails, Adobe Air, iPhone, etc. Con Aptana se facilita en desarrollo integrado de Ajax con las tecnologías emergentes. El líneas generales, así es como se presentan ellos y la verdad es que tiene buena pinta.

Aptana está basado en el conocido entorno de desarrollo Eclipse (IDE = Integrated Development Environment), también Open Source. Pero mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo web, con soporte a HTML, CSS y

Javascript, así como opcionalmente a otras tecnologías mencionadas como PHP, Adobe Air o Ruby on Rails. Aptana Studio está disponible como una aplicación independiente o como plug-in para Eclipse.

En resumen, estas son algunas de las características principales de Aptana Studio:

- ✓ Ayudas visuales para la escritura de scripts en diversos lenguajes, como coloreado y auto escritura del código, ayudas contextuales de referencia a medida que se escribe, etc.
- ✓ Visualización de errores de sintaxis a medida que se escribe.
- ✓ Soporte para hacer FTP a servidores remotos, con herramientas para sincronización.
- ✓ Debug en Firefox (Debug Internet Explorer también con la versión Profesional)
- ✓ Librerías de funciones en Javascript populares en Ajax/Javascript para utilizar en los proyectos.
- ✓ Ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
- ✓ Previsualización de estilos CSS con el editor CSS.
- ✓ Extensible a partir de plug-ins que puede crear Aptana u otras empresas y herramientas para estar al tanto de cualquier nuevo añadido.
- ✓ Extensible por Javascript. Los usuarios pueden escribir scripts para realizar acciones y macros.
- ✓ Los Snippets permiten insertar fragmentos de texto que se utilizan muy a menudo.
- ✓ Y un largo etc...

Todo esto hace de Aptana un programa muy interesante para los desarrolladores del web, gratuito y basado en otro gran sistema Eclipse, lo que garantiza aun más el buen trabajo demostrado hasta ahora por el equipo de Aptana.

### 1.5.8 Frameworks

#### ¿Que es un Frameworks?

En el desarrollo de software, un frameworks es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un frameworks puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a

desarrollar y unir los diferentes componentes de un proyecto. Un frameworks representa una arquitectura de software que modela las relaciones generales de las entidades del dominio.

Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Los Framework son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Framework es un concepto sumamente genérico, se refiere a “ambiente de trabajo, y ejecución”, por ejemplo “.Net” es considerado un “frameworks” para desarrollar aplicaciones (Aplicaciones sobre Windows).

En general los frameworks son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). Un frameworks agrega funcionalidad extendida a un lenguaje de programación; ésta, automatiza muchos de los patrones de programación para orientarlos a un determinado propósito. Un frameworks proporciona una estructura al código y hace que los desarrolladores escriban código mejor, más entendible y manejable. Además hace la programación más fácil, convirtiendo complejas funciones en sencillas instrucciones. Está usualmente escrito en el lenguaje que extiende.

Un frameworks permite separar en capas la aplicación. En general, divide la aplicación en tres capas:

- ✓ La lógica de presentación que administra las interacciones entre el usuario y el software.
- ✓ La Lógica de Datos que permite el acceso a un agente de almacenamiento persistente u otros.
- ✓ La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo a los comandos recibidos desde la presentación.

**Por tanto se puede resumir:**

Un frameworks simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un frameworks proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por

último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

### **Symfony**

Symfony es un completo frameworks diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows. [symfony\_guia\_definitiva, 2007]

### **Características**

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares)
- ✓ Independiente del sistema gestor de bases de datos
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- ✓ Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web

- ✓ Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- ✓ Fácil de extender, lo que permite su integración con las librerías de otros fabricantes

### Automatización de características de proyectos web

Symfony automatiza la mayoría de elementos comunes de los proyectos web, como por ejemplo:

- ✓ La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- ✓ La capa de presentación utiliza plantillas y *layouts* que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del frameworks. Los *helpers* incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones.
- ✓ Los formularios incluyen validación automatizada y relleno automático de datos (*“repopulation”*), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- ✓ Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- ✓ La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- ✓ La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.
- ✓ El sistema de enrutamiento y las URL *limpias* permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.
- ✓ El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones web interactuar más allá de los navegadores.
- ✓ Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.

- ✓ Los plugins, las factorías (patrón de diseño “*Factory*”) y los “*mixin*” permiten realizar extensiones a medida de Symfony.
- ✓ Las interacciones con Ajax son muy fáciles de implementar mediante los *helpers* que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código.

### Entorno de desarrollo y herramientas

Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software:

- ✓ Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- ✓ El frameworks de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas (“*test-driven development*”).
- ✓ La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- ✓ La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- ✓ Es posible realizar cambios “*en caliente*” de la configuración (sin necesidad de reiniciar el servidor).
- ✓ El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

### ¿Por qué usar Symfony?

El principal argumento para decidir si usar o no Symfony es el tamaño del proyecto. Como nuestro proyecto es muy complejo con mucha lógica de negocio, no es recomendable utilizar solo PHP. Para asegurar el mantenimiento y las ampliaciones futuras de la aplicación, es necesario que el código sea

ligero, legible y efectivo. Como se quiere incorporar los últimos avances en interacción con usuarios (como por ejemplo Ajax), se puede escribir cientos de líneas de JavaScript. Se puede desarrollar aplicaciones de forma divertida y muy rápida, no es aconsejable utilizar solo PHP. En todos estos casos, se utilizaría Symfony pues es un frameworks maduro, bien documentado y con una gran comunidad.

### 1.6 Pruebas

*“La prueba es el proceso de un programa con la intención específica de encontrar errores previos a la entrega al usuario final.”* [Juristo - Moreno –Vegas, 2006]

Las pruebas de software son la comprobación del software, de modo general. Esta comprobación se basa fundamentalmente en cuatro aspectos:

- ✓ Demostración (*proof*): manual o semiautomática
- ✓ Inspección manual del código
- ✓ Prueba o ensayo (*testing*): ejecutar y ver resultados
- ✓ Caso de prueba: ensayo individual

El objetivo principal de las pruebas de software es encontrar defectos en el mismo. El éxito de las pruebas radica en encontrar un defecto y el fracaso, en la no detección de defectos existentes. [Collado, 2005]

Para el desarrollo de la interfaz, se realizan pruebas de validación de las interfaces de usuario, en la certificación de cada una de las interfaces de usuario con las que cuentan los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

*“Las pruebas de interfaz de usuario verifican la interacción del usuario con el software. La meta de estas pruebas es asegurar que la interfaz de usuario provee la navegación y el acceso apropiados dentro de la aplicación.”* [Mañas, 2007]

### 1.7 Conclusiones del Capítulo

En el presente capítulo se hizo una descripción de los conceptos asociados a las principales actividades que debe desarrollar un programador y se valoraron las herramientas que utiliza el

programador en el desarrollo de la capa de Presentación de los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos para concluir con la necesidad del desarrollo en esta capa, una interfaz amigable, funcional y rápida, teniendo en cuenta la del empleo eficiente de estas herramientas .

# 2

## Capítulo 2: Descripción de la solución propuesta

### 2.1 Introducción al capítulo

En este capítulo se dará una panorámica de cómo se da respuesta al problema, la solución propuesta, para ello se abordará temas, sin los cuales no hubiese sido posible la realización del sistema. Como detalle principal se debe hablar de la capa de Presentación la cual brinda interfaces que le permiten al usuario intercambiar con el sistema así como la utilización de los componentes que nos brinda la librería Extjs.

Se dará una breve descripción de las interfaces de usuario que contiene la capa de Presentación, cumpliendo los requisitos de forma correcta de ambos módulos en el Sistema Integral para la Gestión de Recursos Humanos.

También tratará sobre las estrategias de integración con AJAX que satisfacen nuestro sistema y la importancia de las mismas para mayor organización y rapidez de la aplicación, así como la estrategia para la captura de errores en los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

### 2.2 Solución Propuesta

Se concentrará en una propuesta de interfaz basada en los estándares de calidad y diseño internacionales que responde con los requerimientos del sistema así como con los requerimientos del usuario además se utilizará las tecnologías más novedosas y que sean aprobadas por el proyecto.

Esto contribuirá en gran medida, al éxito del producto final que garantizara la gestión de información a través de una interfaz funcional, amigable rápida y agradable para el usuario.

### 2.3 Capa de Presentación

La capa de Presentación es la intermediaria entre el usuario y la aplicación, es el frente del sistema, es quien va a comunicarse directamente con el usuario, por tanto debe presentar una buena calidad, entendimiento, debe de cumplir con ciertas normas estándar, esta capa es la encargada de darle la posibilidad al usuario de insertar u obtener información deseada en ella se encuentran las interfaces y clases que nos permiten insertar y validar los datos introducidos por los usuarios.

Para lograr una interfaz con un entorno amigable, sencilla, que permita al usuario una buena y rápida familiarización con la misma. Se utilizaron los componentes que nos brinda librería Extjs, la cual ha sido integrada al Framework Symfony para proveer una interfaz más funcional.

La capa de Presentación esta compuesta por Plantillas, Helpers, Layouts y la librería Extjs.

Una **plantilla** es un archivo que está ubicado en el directorio *templates/del módulo*, y su nombre está compuesto por el nombre de la acción y el resultado de la misma, solo deben contener código de presentación, así que hay que procurar mantener la menor cantidad de código PHP en ellas como sea posible. El contenido de las plantillas está formado por código HTML y algo de código PHP sencillo.

Los **helpers** son funciones de PHP que devuelven código HTML y que se utilizan en las plantillas, a veces, los helpers solamente se utilizan para ahorrar tiempo, agrupando en una sola instrucción pequeños trozos de código utilizados habitualmente en las plantillas además que facilitan la creación de las plantillas y producen el mejor código HTML posible en lo que se refiere al rendimiento y a la accesibilidad. Aunque se puede usar HTML normal y corriente, los helpers normalmente son más rápidos de escribir. Una pequeña lista de los helpers que más se utilizan y muestra también el código HTML que generan.

Los helpers utilizados en la sección **<head>** obtienen información del objeto respuesta y en la configuración de la vista. La etiqueta **<body>** muestra el resultado de la plantilla.

Los **Layouts** en las páginas se encuentran en otro lugar de la aplicación, un archivo llamado *layout.php* que contiene el layout de la página. Este archivo, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro

punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de diseño llamado “decorator”

La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones. [symfony\_guia\_definitiva, 2007]

La **librería Extjs** es una librería construida con JavaScript que proporciona una interfaz a las famosas librerías de Yahoo!, jQuery y Prototype + Scriptaculous, su potencia radica en la rica colección de componentes para el diseño de GUI's del lado del cliente haciendo uso extensivo de Ajax.

Entre los componentes que esta librería ofrece se encuentran cuadros de diálogo, menús, tablas editables, layouts, paneles, pestañas y todo lo necesario para construir atractivos desarrollo de las interfaces de usuario.

### En la Capa de Presentación:

- ✓ Los formularios incluyen validación automatizada y relleno automático de datos (“repopulation”), lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.
- ✓ Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.
- ✓ La gestión de la caché reduce el ancho de banda utilizado y la carga del servidor.
- ✓ La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.
- ✓ El sistema de enrutamiento y las URL limpias permiten considerar a las direcciones de las páginas como parte de la interfaz, además de estar optimizadas para los buscadores.
- ✓ Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.
- ✓ Los plugins, las factorías (patrón de diseño “Factory”) y los “mixin” permiten realizar extensiones a medida de Symfony.

- ✓ Las interacciones con Ajax son muy fáciles de implementar mediante los helpers que permiten encapsular los efectos JavaScript compatibles con todos los navegadores en una única línea de código.

Existen numerosas herramientas y utilidades para manipular la capa correspondiente a la presentación. Las plantillas se pueden construir en pocos segundos, gracias al uso de los helpers. Los layouts, los elementos parciales, los componentes y los slots de componentes permiten aplicar los conceptos de modularidad y reutilización de componentes. La configuración de la vista aprovecha la velocidad de YAML para manejar la mayoría de cabeceras de las páginas. La configuración en cascada evita tener que definir todas las opciones para cada vista.

### 2.3.1 Breve descripción de las Interfaces que contiene la capa de Presentación

La mayoría de los prototipos de Interfaces de Usuario están basados en formularios, formularios que se encuentran en las plantillas, gracias a los *helpers* de formularios que incluye Symfony y a sus opciones avanzadas, simplifican enormemente el desarrollo de la Interfaz. Los archivos de validación, los *helpers* de validación y la opción de volver a mostrar los datos en un formulario, permiten reducir el esfuerzo necesario para crear un control estricto de los formularios que sea robusto y a la vez fácil de utilizar por parte de los usuarios.

Ahora se hace una breve descripción de las interfaces de usuario en los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

#### 2.3.1.1 Módulo “Selección al Ingreso”

El módulo “**Selección al Ingreso**” esta conformado por 15 Casos de Uso, a continuación se presenta la Interfaz de Usuario para cada uno de ellos:

##### ➤ *CUEstablecerNecesidadesdeConvocatoria:*

Definir si es necesario poner en convocatoria una plaza que se encuentre vacante. El sistema muestra todas las plazas que se encuentran vacantes en una unidad determinada y las fichas que se

encuentran en el estado de Bolsa que están asociadas a la misma. Y le permite al actor escoger que ficha será asignada a dicha plaza.

### Plazas Vacantes



**Nota:** Se selecciona la unidad, a continuación se muestran las plazas vacantes para dicha unidad en el campo Plazas Vacantes, a continuación se selecciona una plaza y en el campo fichas se muestra todas las fichas que se encuentran en estado bolsa que están optando por dicha plaza.



Y si no hay Plazas Vacantes...

#### ➤ *CUVerificarPlazasVacantes:*

Muestra las plazas que estén en el estado Vacante en la plantilla de una unidad, después que sea llamado del caso de uso Culminar Ficha o Establecer Necesidades de Convocatoria.



Luego de esto se verifica si el candidato es desertor o si ya ha comenzado algún proceso.



Si el candidato se le ha realizado el proceso anteriormente muestra la fecha del último proceso y continúa el proceso.

### ➤ *CUCrearFicha:*

Creación de la ficha y recogida de datos personales, de contacto, de formación y otros datos del candidato. Pero antes debe verificar si el candidato ha sido desertor o si se le ha realizado procesos antes.

#### Datos personales



**Datos Personales** | Datos de Contacto | Datos de Formación | Complementarios

**Datos de Identidad**

Carnet de Identidad: 84060506985

Primer Nombre:

Segundo Nombre:

Primer Apellido:

Segundo Apellido:



**Datos Personales**

Fecha de Nacimiento: 05/06/1984

Provincia de Nacimiento: Seleccione...

Municipio de Nacimiento: Seleccione...

Sexo: Masculino

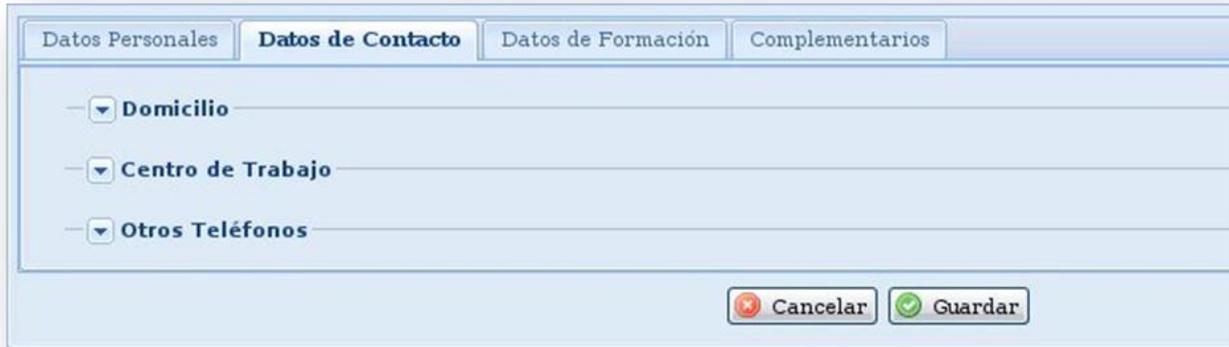
Estado Civil: Seleccione...

Cantidad de Hijos: 0

Raza: Seleccione...

Cancelar Guardar

Datos del Contacto

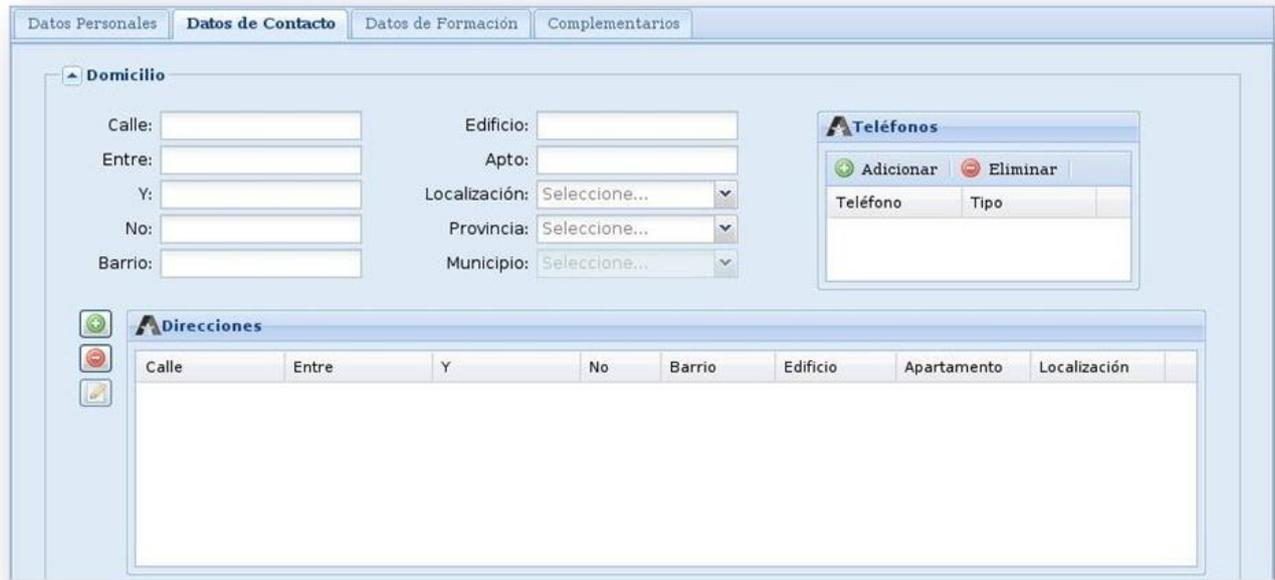


Datos Personales **Datos de Contacto** Datos de Formación Complementarios

Domicilio  
 Centro de Trabajo  
 Otros Teléfonos

Cancelar Guardar

Datos del Domicilio



Datos Personales **Datos de Contacto** Datos de Formación Complementarios

Domicilio

Calle:  Edificio:   
 Entre:  Apto:   
 Y:  Localización: Seleccione...  
 No:  Provincia: Seleccione...  
 Barrio:  Municipio: Seleccione...

**Teléfonos**

Teléfono	Tipo

**Direcciones**

Calle	Entre	Y	No	Barrio	Edificio	Apartamento	Localización

Datos del Centro de Trabajo

Centro de Trabajo

Nombre:  Edificio:

Calle:  Apto:

Entre:  Barrio:

Y:  Provincia:

No:  Municipio:

**Telefonos**

Adicionar Eliminar

Telefono	Tipo

**Direcciones**

Centro de Trabajo	Calle	Entre	Y	No	Barrio	Edificio	Apartamento

Otros Teléfonos

Otros Teléfonos

Número:

Tipo:

Localización:

**Direcciones**

Número	Tipo	Localización

### Información sobre la última escolaridad de la que tenga titulación



**Datos de Formación**

**Escolaridad Vencida**

Escolaridad vencida:  Provincia:

Centro de enseñanza:  Municipio:

**De su titulación**

Tomo:

Folio:

Fecha:

**De quién lo otorga**

Nombre:

Apellidos:

Cargo:

### Otros Datos



**Complementarios**

**Cargo solicitado**

Unidad:  Plaza:   Forma de presentación:

Sanciones

Trámites migratorios

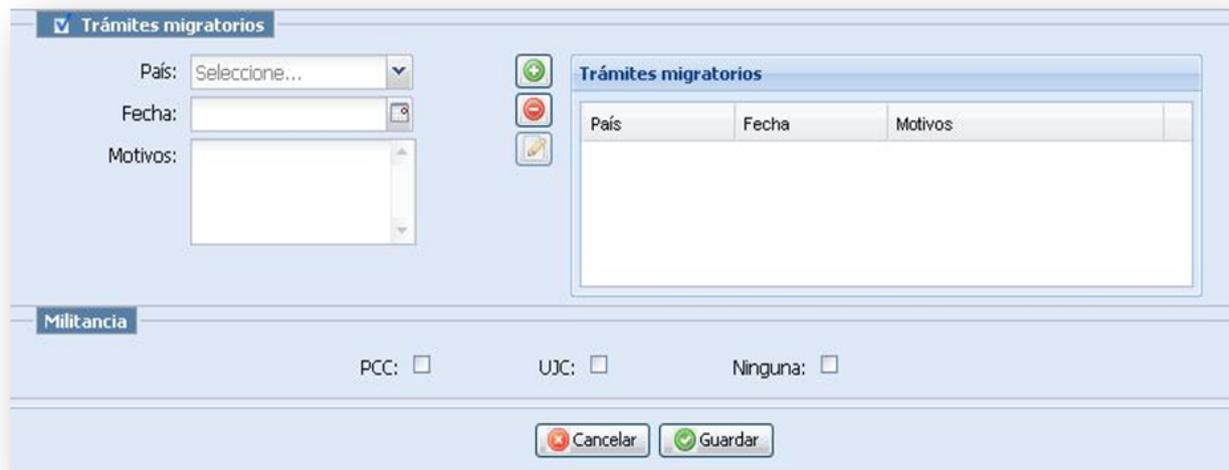
**Militancia**

PCC:  UJC:  Ninguna:

**Nota:** En este campo el usuario al ejecutar el evento doble click aparece un pop-up donde busca la plaza por la que esta optando el candidato.



**Nota:** Si se marca Sanciones, en caso de que exista alguna sanción lo cual trae consigo habilitar los campos ¿Qué sanción?, En Donde y Fecha, además de habilitar los botones de adicionar, modificar y eliminar las sanciones de la tabla se sanciones.



**Nota:** Si se marca en caso de que el candidato haya realizado trámites migratorios lo que trae consigo habilitar los campos de Fecha, País, Motivos.

Los 3 botones son uno para adicionar a la tabla la sanción que se recoja en los campos referentes a la sanción . El segundo es para editar la sanción que se encuentre seleccionada en la tabla se sanciones, al pulsar este botón se deben cargar los datos de la tabla en los campos correspondientes

de la sanción . El tercer botón se utiliza para eliminar la sanción que se encuentre marcada en la tabla se sanciones .

### ➤ *CUResultadosPrimeraEntrevista:*

Guardar el resultado de la primera entrevista que se le realiza al candidato. El psicólogo registra en caso necesario alguna observación realizada al candidato y esta se plasma en la ficha, además de dar un resultado de aprobado, no aprobado; lo cual cambia el estado de la misma.



No Fichas	Carne de Identidad	Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido
200805001	88021911131	odelkis		Rivera	Ortiz

Reporte de las Fichas

Pág 1 de 1 Fichas 1 - 1 de 1

**Nota:** Se carga los datos de todas las fichas que se encuentra en el estado de Creada. En el momento de darle doble click a una se debe cargar la siguiente pantalla donde se muestra la información mas detallada del candidato.

**Información**

	CI: <input type="text"/>	No Ficha: <input type="text"/>
	Nombre(s): <input type="text"/>	Fecha Inicio: <input type="text"/>
	Apellidos: <input type="text"/>	Cargo: <input type="text"/>

¿ Se le continuará el Proceso ?       Si       No

Observaciones :

**Nota:** en el campo Información del candidato se cada la información referente al candidato como es nombre y sus apellidos, la foto si la tiene, número de carnet de identidad, número de ficha y fecha de inicio del proceso y cargo para el que se esta procesando. Luego que verifica si el cargo por el que se esta procesando necesita psicométrico o no y guarda observación.



Si no existe ninguna ficha pendientes a los resultados de la Primera Entrevista, El sistema regresa al menú principal.

➤ ***CUResultadosAnálisisPsicologico:***

Guardar el resultado del análisis psicológico que se le realizó al candidato además de cambiarle el estado a la ficha.

No Fichas	Carne de Identidad	Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido
200805001	88021911131	odelkis		Rivera	Ortiz

Reporte de las Fichas

Pág 1 de 1

Fichas 1 - 1 de 1

**Nota:** Se carga los datos de todas las fichas que se encuentra en el estado de psicométrico. En el momento de darle doble click a una se debe cargar la siguiente pantalla donde se muestra la información mas detallada del candidato.

**Información**



CI:	<input type="text"/>	No Ficha:	<input type="text"/>
Nombre(s):	<input type="text"/>	Fecha Inicio:	<input type="text"/>
Apellidos:	<input type="text"/>	Cargo:	<input type="text"/>

¿ Se le continuará el Proceso ?       Si       No

Observaciones :

**Nota:** en el campo Información del candidato se carga la información referente al candidato como es nombre y sus apellidos, la foto si la tiene, número de carnet de identidad, número de ficha y fecha de inicio del proceso y cargo para el que se esta procesando. Luego que verifica si el cargo por el que se esta procesando necesita psicométrico o no y guarda observación.



### ➤ *CUResultadosVerificacion:*

Guarda el resultado la verificación que se le realiza al candidato por los órganos de la UCESA. Los resultados de la misma solamente es aprobado o desaprobado.



A screenshot of a web application window titled "Reporte de las Fichas". It displays a table with the following data:

No Fichas	Carne de Identidad	Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido
200805001	88021911131	odelkis		Rivera	Ortiz

At the bottom of the window, there is a pagination control showing "Pág 1 de 1" and a status bar indicating "Fichas 1 - 1 de 1".

**Nota:** Se carga los datos de todas las fichas que se encuentra en el estado de Pendiente a Verificación. En el momento de darle doble click a una se debe cargar la siguiente pantalla donde se muestra la información mas detallada del candidato.

**Información**

	CI: <input type="text"/>	No Ficha: <input type="text"/>
	Nombre(s): <input type="text"/>	Fecha Inicio: <input type="text"/>
	Apellidos: <input type="text"/>	Cargo: <input type="text"/>

¿ Se le continuará el Proceso ?       Si       No

Observaciones :

**Nota:** en el campo Información del candidato se carga la información referente al candidato como es nombre y sus apellidos, la foto si la tiene, número de carnet de identidad, número de ficha y fecha de inicio del proceso y cargo para el que se esta procesando.

**Información** ✕

 Se registraron correctamente los datos

### ➤ CUPendientes a Aceptación:

Buscar las fichas que se encuentran en el estado de Pendiente a Aceptación y confirmar la aceptación de la plaza o marcar al candidato como desertor. El psicólogo consulta las fichas que se encuentran pendientes a aceptación, además de confirmar la aceptación y asignación de la plaza, además de generar la carta de aprobación.



The screenshot shows a window titled "Pendientes a Aceptación" with a sub-window titled "Fichas". The "Fichas" window contains a table with the following data:

Nombre	No CI
odelkis Rivera Ortiz	88021911131

Below the table is a "Cancelar" button.

**Nota 1:** En el panel izquierdo se cargan las fichas que estén en el estado de Pendientes a Aceptación, con los siguientes datos número de carnet de identidad, nombre y apellidos del candidato, en caso de que no marque ninguno la ficha se mantiene en el estado actual.



The screenshot shows a form titled "Información" with a profile picture placeholder on the left. The form contains the following fields:

- CI:
- No Ficha:
- Nombre(s):
- Fecha Inicio:
- Apellidos:
- Cargo:

Y cuando no existen Fichas disponibles:



➤ **CUPlanificarDiasPsicometrico:**

Planificar los días que se realizara las pruebas del psicométrico.



**Nota:** En el campo días planificados se cargara los días planificados hasta la fecha y se van incorporando los nuevos días planificados con la cantidad de personas. Los botones , ,  son para adicionar un nuevo día, para eliminar un día seleccionado el la lista de días planificados y para editar un día ya creado sea cambiándole la cantidad de personas o la fecha del mismo además del lugar donde se va a realizar la prueba psicométrica.

### ➤ *CUPlanificarPsicometrico:*

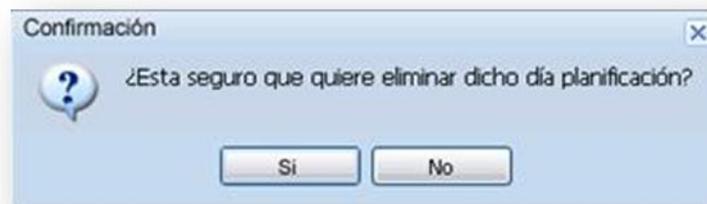
Planificar que candidatos van a asistir a un determinado día de psicométrico. El psicólogo debe escoger un día de los que tiene ya planificados para la realización del psicométrico y comenzar a asignarle candidatos, para que sean atendidos el día planificado



**Nota:** En el campo fichas se encuentran las fichas que están en estado **Presentada** ordenadas por fecha de forma ascendente, y en el campo Fichas para dd/mm/aaaa se van cargando las fichas que han sido seleccionadas para realizarle el psicométrico el día seleccionado.

### ➤ *CUCancelarPsicometrico:*

Cancelar la planificación de las pruebas psicométricas de un día determinado. el psicólogo cancela un día determinado de psicométrico debido a algún problema imprevisto a la hora de planificar, siempre teniendo en cuenta que ese día este previamente planificado.



➤ ***CURegistrarAsistenciaalPsicometrico:***

Registrar los candidatos que asistieron realmente a la prueba psicométrica. El psicólogo registra cuales de todos los candidatos que se les planifico el psicométrico para una fecha determinada asistieron realmente dicha prueba.



**Nota:** Aquí se mostrarán las fichas planificadas para la fecha seleccionada, de ellas se debe mostrar el número de la ficha, el nombre y apellidos del candidato y en número de carnet de identidad. Además por cada ficha se adiciona un checkbox para marca o no que a esta ficha se le realizo la prueba psicométrica.

### ➤ *CUCrearAgenda:*

Se crea una agenda para la planificación de las fichas las cuales fueron aprobadas en el proceso de selección al ingreso, los trabajadores a los cuales se les realizarán un nuevo contrato o a los que se les va a realizar un anexo a su contrato. Todos estos serán presentados a la reunión de los órganos colegiados para su aprobación o no. El usuario este autenticado como Secretario de Órganos Colegiados.

Si ya existe Agenda:



Si no:

**Crear Agenda**

**Fichas**

Nombre	Cargo

Pág 1 de 1

Fecha:

**Fichas para dd/mm/aaaa**

Nombre	Tipo Contrato	Cargo

**Crear Agenda**

Comisión **Tipo**      Fecha para la realización de la Comisión:

**Pedidos a Nuevos Contratos**

Trabajadores que solicitan nuevos contratos

Trabajadores para la Comisión de Organos Colegiados



**Nota:** Tipo carga el tipo de comisión según el usuario autenticado (Comisión de cuadro y Comisión representativa)

**Nota 1:** El TipoComision es referente a la comisión que correspondiente ya sea Comisión de Cuadro y Comisión Representativa y Unidad carga el nombre de la unidad a la cual se le esta realizando la comisión de cuadros

**Nota 2:** Cuando sale la ventana de la información del candidato sale un campo donde se recoge el tipo de contrato, esto se realiza ejecutando la acción doble click sobre el campo se levanta un pop-up con todos lo tipos de contrato para que el usuario seleccione el que desee.

**Nota 3:** En caso de que quiera quitar ( - ) una plaza de la planificación de la agenda, se debe eliminar el tipo de contrato almacenado en la plaza asignada a dicho candidato.

**Nota 4:** Estas tres pantallas forman una sola separadas por pestañas. Manteniendo el campo fecha que es común para cualquier planificación.

### ➤ *CUAprobacionCandidatos:*

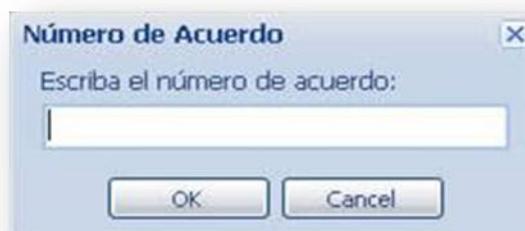
Aprobación de las personas ya sea trabajadores o candidatos que se encuentran planificados en la agenda. El actor puede aprobar o posponer las personas (trabajadores o candidatos) que se encuentran en la agenda de la reunión que realizan los órganos colegiados. Tener en cuenta que ya exista una agenda.



Número de Carnet de Identidad	Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido	Aprobado	Posteado
85103018735	Yilixy	Martinez	Garcia		<input type="radio"/>	<input type="radio"/>

**Tipo:** Es el tipo de comisión que esta seccionando.

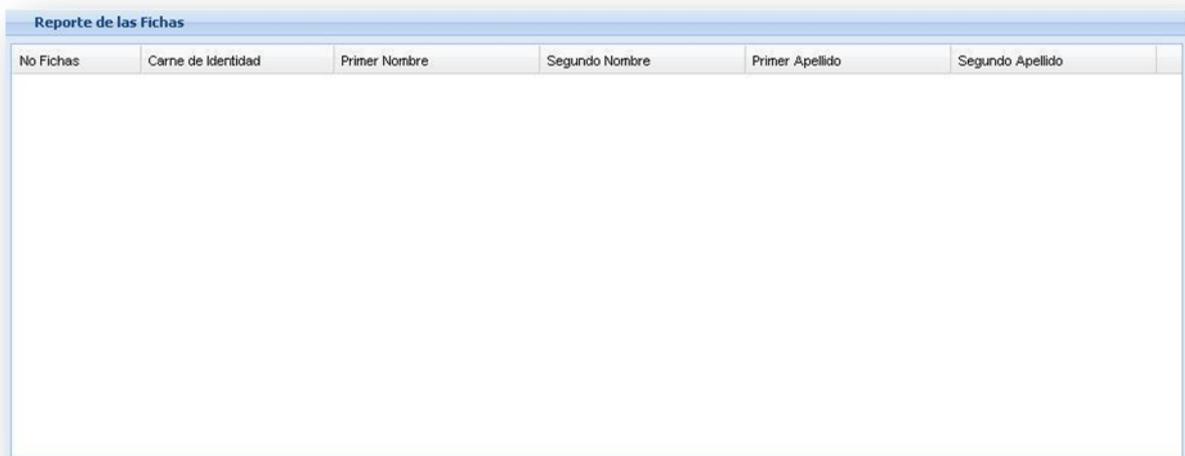
**Nota:** En cada una de las secciones se cargan también la misma información que en la primera, incluyendo los radiobutton para aprobar o no a la persona.



Para la comisión de Cuadros: se generan la Resolución del nombramiento. Genera el documento de salida de órganos colegiados según el órgano colegiado que se este llevando a cabo, en caso de que sea comisión representativa se genera Acuerdo de Nombramiento, y en caso de comisión de cuadros se genera el documento Resolución del Nombramiento.

### ➤ *CUCulminarFicha:*

Dar por concluido el proceso de selección y terminar la recogida de los últimos datos de la ficha.



No Fichas	Carne de Identidad	Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido

**Nota:** Se carga los datos de todas las fichas que se encuentra en el estado de Verificación. En el momento de darle doble click a una se debe cargar la pantalla donde se muestra la información mas detallada del candidato.

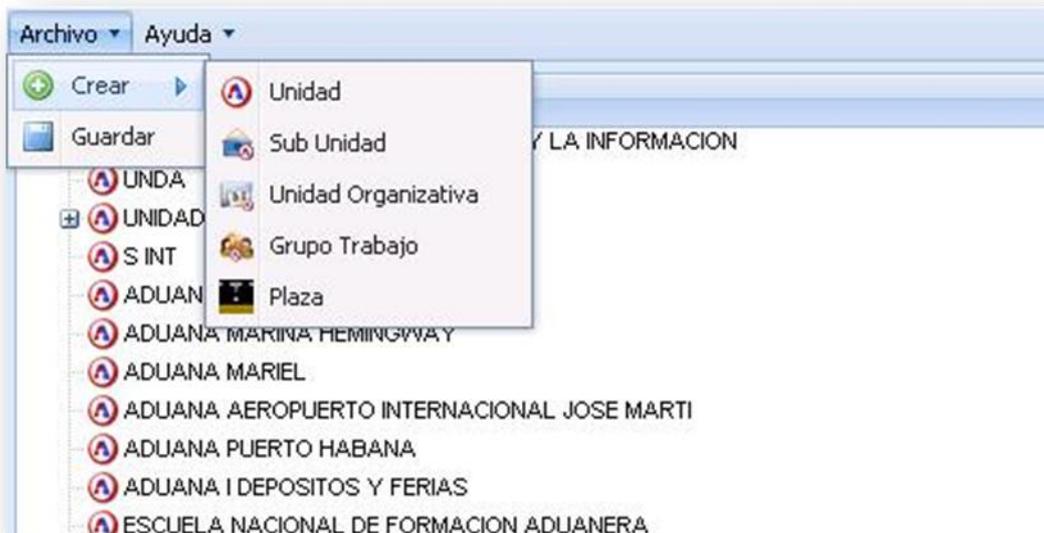
### ➤ *CUAsignarPlaza:*

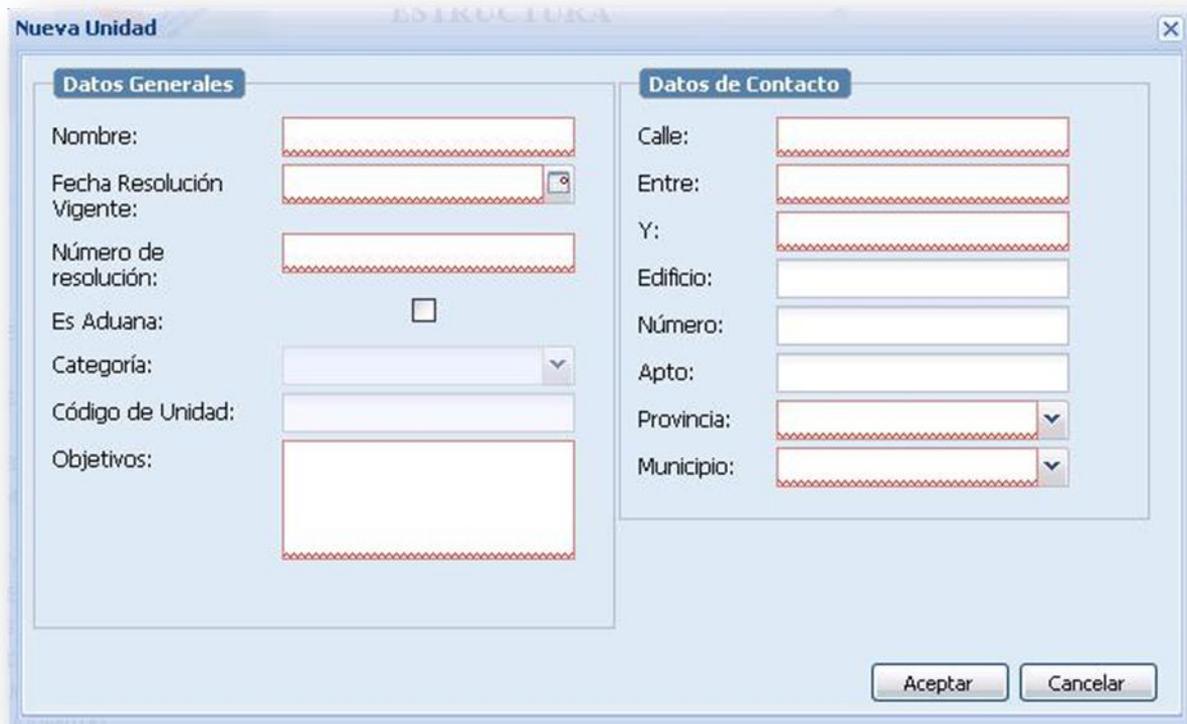
Marca una plaza vacante en el sistema como Reservada para un candidato determinado. Se le asigna una plaza previamente seleccionada por el psicólogo al candidato modificándosele el estado de la misma a Reservado, y ejecutándose el caso de uso Generar Carta Aprobación.



### ➤ *CUCrearUnidad:*

Creación de una unidad y se definen un conjunto de datos de la unidad, estos son: el nombre, siglas, objetivos, dirección, número resolución y fecha de dicha resolución. Esta información se guarda en el sistema. En caso de que se cree una aduana debe incluirse también código de la unidad y la categoría, el sistema debe proponer una estructura según la categoría escogida, en caso contrario el sistema ejecuta el caso de uso confeccionar estructura.

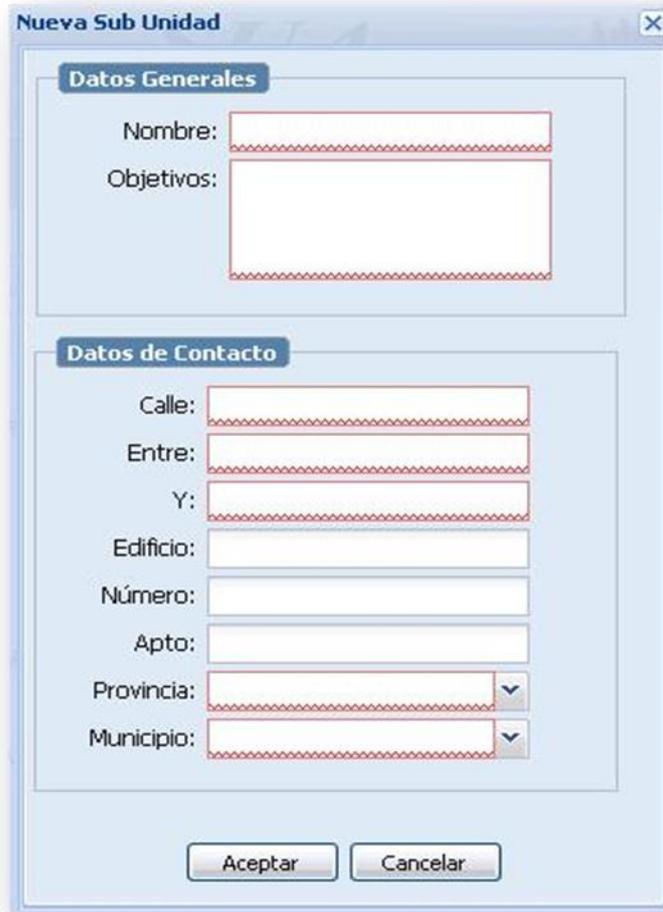




**Nota:** Al marcar el checkbox con el nombre Es Aduana deben ponerse visibles los campos categoría y código del unidad.

➤ ***CUCrearSubUnidad:***

El objetivo de este caso de uso es crear una sub-unidad para subordinarla a alguna unidad de la Aduana General de la República (AGR). El actor selecciona un elemento de la estructura y después la opción de crear Sub-Unidad. Luego registra los datos para crear una sub-unidad, estos datos son: el nombre y los objetivos. El sistema debe validar que la para la opción seleccionada se pueda crear una Sub-Unidad y en caso de que ocurra algún error notificar con un mensaje de error. En caso de que todo este correcto debe mostrar la sub-unidad subordinada a una unidad.



**Nueva Sub Unidad**

**Datos Generales**

Nombre:

Objetivos:

**Datos de Contacto**

Calle:

Entre:

Y:

Edificio:

Número:

Apto:

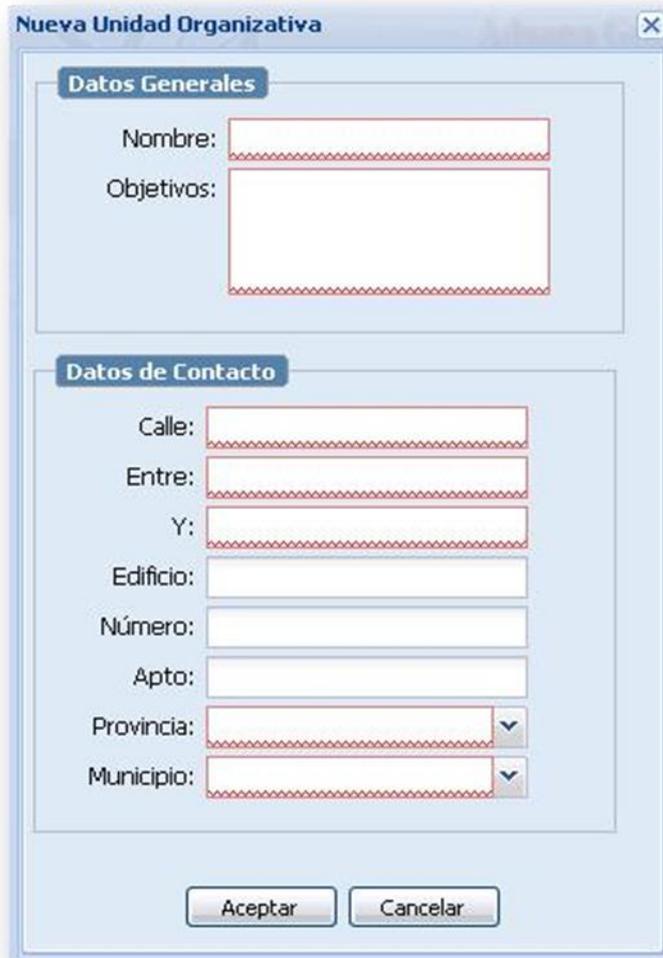
Provincia:

Municipio:

Aceptar Cancelar

### ➤ *CUCrearUnidadOrganizativa:*

El objetivo de este caso de uso es crear una Unidad Organizativa y subordinarla a alguna unidad o sub-unidad ya creada en la estructura de la Aduana General de la República (AGR). El actor selecciona un elemento de la estructura y luego la opción Crear Unidad Organizativa, registra los datos necesarios para crear una unidad organizativa, estos datos son: el nombre y objetivos. El sistema debe validar que el elemento seleccionado de la estructura se corresponda con una Unidad o una Sub-Unidad para poder realizar la subordinación de forma correcta, en caso contrario debe mostrar una notificación de error.



**Nueva Unidad Organizativa**

**Datos Generales**

Nombre:

Objetivos:

**Datos de Contacto**

Calle:

Entre:

Y:

Edificio:

Número:

Apto:

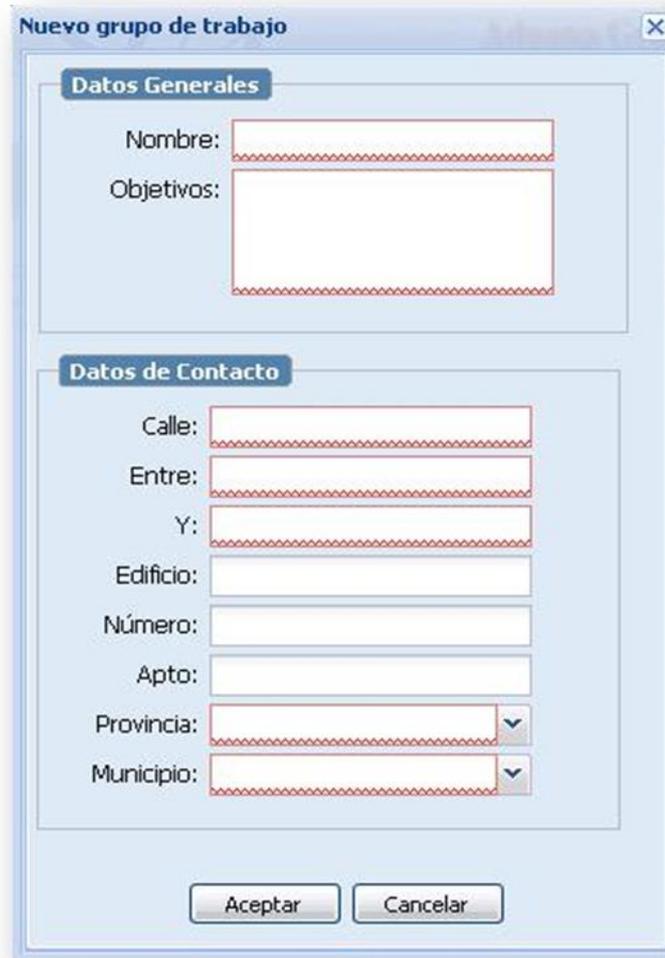
Provincia:

Municipio:

Aceptar Cancelar

➤ ***CUCrearGrupo:***

El objetivo de este caso de uso es crear un grupo de trabajo, definiendo un conjunto de datos y subordinándolo a alguna unidad, sub-unidad o unidad organizativa ya creadas. El actor selecciona una unidad, sub-unidad o unidad organizativa y luego la opción Crear Grupo de Trabajo, define los datos necesarios para crear un grupo, los mismos son: el nombre y los objetivos. El sistema subordina el Grupo de Trabajo a la Unidad, Sub-Unidad o Unidad Organizativa previamente seleccionada.



**Nuevo grupo de trabajo**

**Datos Generales**

Nombre:

Objetivos:

**Datos de Contacto**

Calle:

Entre:

Y:

Edificio:

Número:

Apto:

Provincia:

Municipio:

Aceptar Cancelar

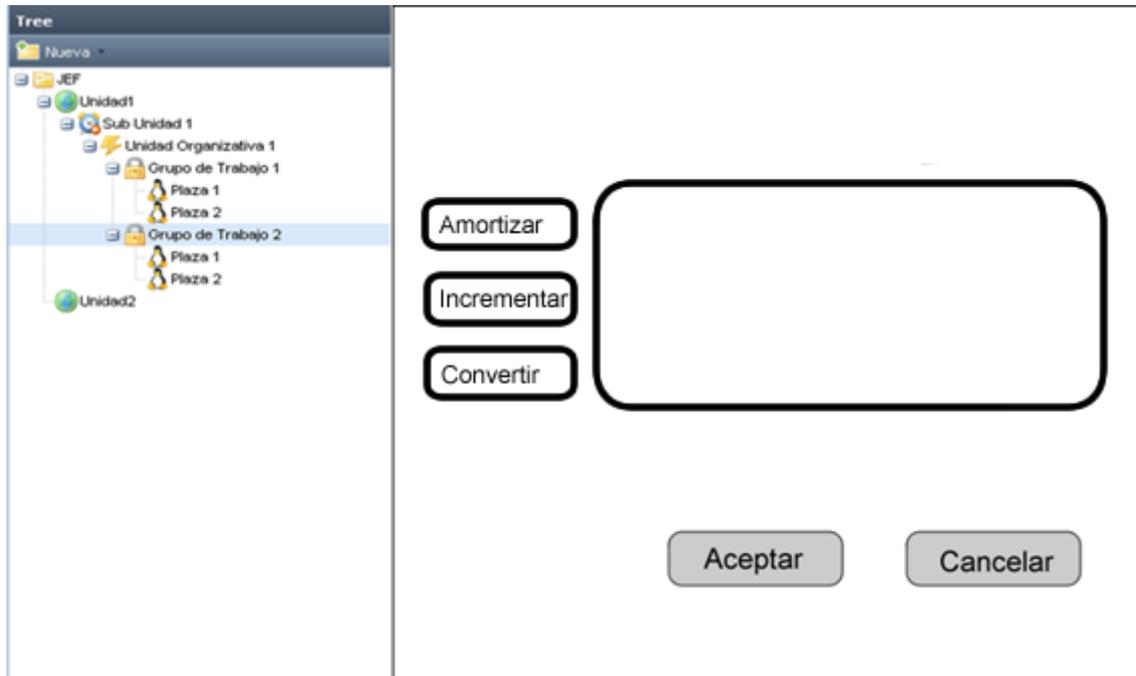
➤ **CUCrearPlaza:**

El objetivo de este Caso de Uso es crear una plaza y subordinarla a un grupo de trabajo. El actor selecciona un grupo de trabajo que se encuentra subordinado a una Unidad, Sub-Unidad o Unidad Organizativa, luego selecciona la opción crear Plaza de Trabajo y define los datos necesarios para crear una nueva plaza. El sistema guarda la información y subordina la plaza al grupo seleccionado.



➤ ***CUGestionarPlaza:***

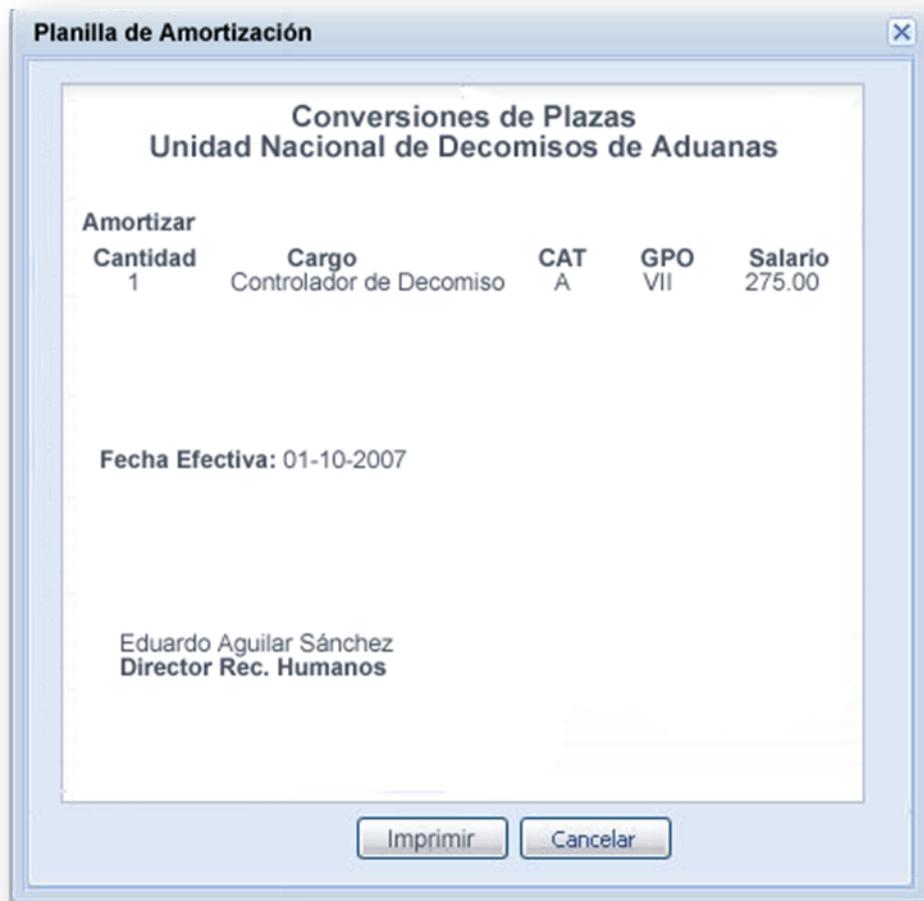
El objetivo de este caso de uso es realizar las amortizaciones, las conversiones y los incrementos de plazas de un Grupo de Trabajo, esto consiste en eliminar la plaza del grupo, convertir una plaza en otra en un determinado grupo y crear una nueva plaza en un grupo de trabajo. Al terminar de gestionar las plazas se imprime la planilla de Conversiones de Plazas. El actor selecciona una plaza dentro de un grupo de trabajo a la cual es la que le va a realizar una de las tres acciones correspondientes a la plaza. Se pueden realizar más de una acción de cada tipo y al concluir de amortizar, de convertir o incrementar las plazas se imprime la plantilla de Conversiones de Plazas.



**Nota:** Comprueba que se halla seleccionado un grupo de trabajo para poder realizar el incremento, en caso contrario muestra un mensaje de “operación no permitida” y ejecuta el caso de uso Crear Plaza.

➤ ***CURealizarAmortizacion:***

El objetivo de este caso de uso es realizar las amortizaciones de plazas de un Grupo de Trabajo, esto consiste en eliminar la plaza del grupo y generar una planilla de Amortización de Plaza. El actor selecciona una plaza dentro de un grupo de trabajo para realizar la amortización, esta es una opción que brinda el sistema. Una vez que se ha terminado de realizar la amortización se genera una planilla de Amortización de Plazas que será impresa por el Actor.



**Planilla de Amortización**

**Conversiones de Plazas**  
**Unidad Nacional de Decomisos de Aduanas**

Amortizar	Cantidad	Cargo	CAT	GPO	Salario
	1	Controlador de Decomiso	A	VII	275.00

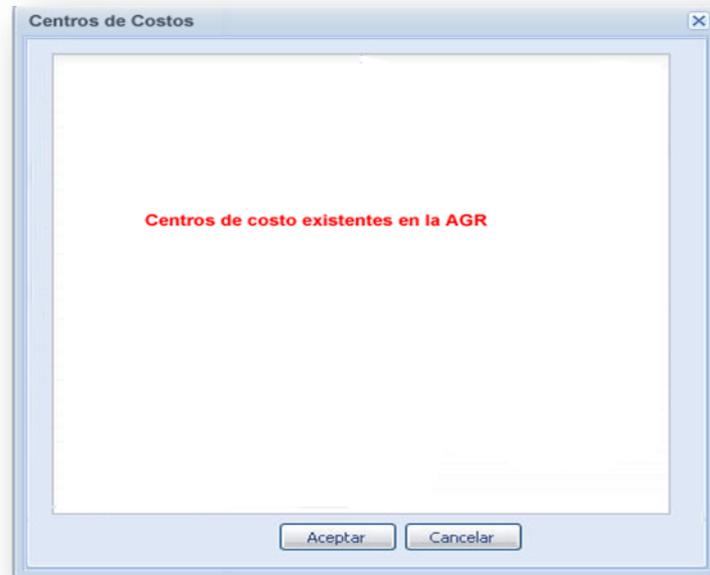
Fecha Efectiva: 01-10-2007

Eduardo Aguilar Sánchez  
 Director Rec. Humanos

Imprimir Cancelar

➤ **CUAsignarPatrimonio:**

Este caso de uso tiene como objetivo asignarle patrimonio a una aduana creada con anterioridad. El actor selecciona una unidad de la estructura de la Aduana General de la República (AGR) para asignarle patrimonio. Esto implica que se registre el centro de costo al que pertenece la unidad, especificando si es propio o no.



➤ ***CUAsignarPersonalidadJurídica:***

El objetivo de este caso de uso es asignar personalidad jurídica a una unidad de la Aduana General de la República (AGR) previamente creada. Con este caso de uso el actor tiene como objetivo asignarle personalidad jurídica a una unidad creada en la AGR. Para esto se registra que la unidad tiene personalidad jurídica y la fecha de la resolución del Ministerio de Economía y Planificación (MEP) y el número de dicha resolución.



➤ **CU Realizar Conversiones de Plazas:**

El objetivo de este caso de uso es Convertir una plaza por otra. Esto consiste en cambiar el cargo de plaza por otro y generar una planilla de Conversión de plaza donde se muestran las características de la plaza antigua y de la nueva. El actor selecciona la plaza que quiere convertir y luego especifica las nuevas características de la plaza, estas son: el cargo. El sistema guarda los cambios realizados y genera una planilla de Conversión de Plazas que será impresa por el usuario.



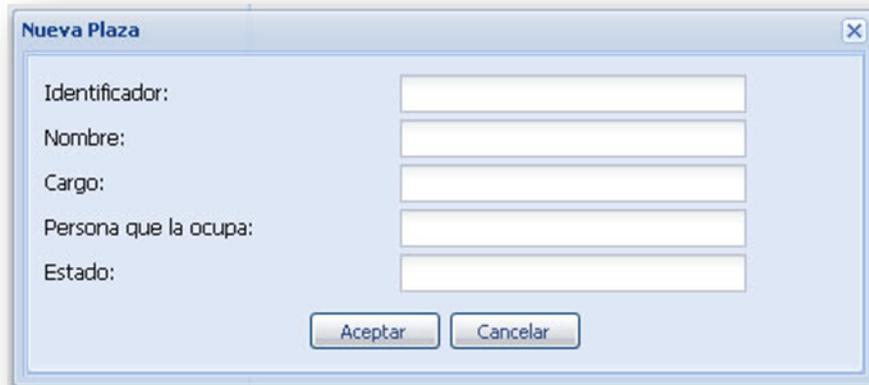
The image shows a software dialog box titled "Conversión de plaza". It has a standard Windows-style title bar with a close button (X) in the top right corner. The dialog contains five text input fields, each preceded by a label: "Identificador:", "Nombre:", "Cargo:", "Persona que la ocupa:", and "Estado:". Below these fields are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Nota: en el evento doble click del cargo se abre una ventana emergente con todos los cargos de la aduana para escoger el nuevo cargo



➤ ***CURealizarIncrementos:***

Con este caso de uso el actor persigue el objetivo de realizar incrementos de nuevas plazas a un determinado Grupo de Trabajo subordinado a una Unidad, sub-unidad o Unidad Organizativa. El actor selecciona un Grupo de Trabajo, luego la opción realizar Incrementos de Plazas. Registra un conjunto de datos necesarios para registrar el incremento, estos datos son: el cargo de la nueva plaza, la categoría ocupacional del cargo, el grupo escala y el salario devengado por ese cargo. Finalmente se genera una planilla de Incremento de Plazas para que el usuario la imprima.



Nota: en el evento doble click del cargo se abre una ventana emergente con todos los cargos de la aduana para escoger el nuevo cargo.

### ➤ *CUEditarEstructura:*

Este caso de uso es una agrupación de casos de usos para gestionar las operaciones a realizar sobre la estructura. Este caso de usos se utiliza para agrupar las operaciones de creación de componentes de la estructura que se pueden realizar, estas operaciones se pueden realizar fundamentalmente en dos momentos, uno para crear nuevos componentes de una estructura (sub-unidad, unidad organizativa, grupo de trabajo, plaza) subordinadas a una unidad que no posee categoría; el otro momento en que se pueden realizar las operaciones es cuando se crea una unidad que posee categoría y se realizan modificaciones a la estructura propuesta por el sistema . Las operaciones están agrupadas en casos de uso independientes y pueden ser las siguientes: ***Crear Unidad Organizativa, Crear Sub-Unidad, Crear Grupo de Trabajo, Crear Plaza.***

## 2.4 Integración con AJAX

Las aplicaciones de la denominada Web incluyen numerosas interacciones en el lado del cliente, efectos visuales complejos y comunicaciones asíncronas con los servidores. Todo lo anterior se realiza con JavaScript, pero programarlo manualmente es una tarea tediosa y que requiere de mucho tiempo para corregir los posibles errores. Afortunadamente, Symfony incluye una serie de *helpers* que automatizan muchos de los usos comunes de JavaScript en las plantillas. La mayoría de comportamientos en el lado del cliente se pueden programar sin necesidad de escribir ni una sola línea

de JavaScript. Los programadores solo tienen que ocuparse del efecto que quieren incluir y Symfony se encarga de lidiar con la sintaxis necesaria y con las posibles incompatibilidades entre navegadores.

Ahora se describen las herramientas proporcionadas por Symfony para facilitar la programación en el lado del cliente:

- ✓ Los *helpers* básicos de JavaScript producen etiquetas `<script>` válidas según los estándares XHTML, para actualizar elementos DOM (Document Object Model) o para ejecutar un script mediante un enlace.
- ✓ Prototype es una librería de JavaScript completamente integrada en Symfony y que simplifica el desarrollo de scripts mediante la definición de nuevas funciones y métodos de JavaScript.
- ✓ Los *helpers* de Ajax permiten al usuario actualizar partes de la página web pinchando sobre un enlace, enviando un formulario o modificando un elemento de formulario.
- ✓ Todos estos *helpers* disponen de múltiples opciones que proporcionan una mayor flexibilidad, sobre todo mediante el uso de las funciones de tipo *callback*.
- ✓ Script.aculo.us es otra librería de JavaScript que también está integrada en Symfony y que añade efectos visuales dinámicos que permiten mejorar la interfaz y la experiencia de usuario.
- ✓ JSON (JavaScript Object Notation) es un estándar utilizado para que un script de cliente se comunique con un servidor.

Las aplicaciones Symfony también permiten definir interacciones complejas en el lado del cliente, combinando todos los elementos anteriores. Mediante una sola línea de código PHP (la llamada al helper de Symfony) es posible incluir las opciones de autocompletado, arrastrar y soltar, listas ordenables dinámicamente y texto editable. [symfony\_guia\_definitiva, 2007]

### 2.4.1 Helpers básicos de Javascript

Javascript siempre se había considerado como poco útil en el desarrollo de aplicaciones web profesionales debido a sus problemas de incompatibilidad entre distintos navegadores. Hoy en día, se han resuelto la mayoría de incompatibilidades y se han creado librerías muy completas que permiten programar interacciones complejas de Javascript sin necesidad de programar cientos de líneas de código y sin perder cientos de horas corrigiendo problemas. El avance más popular se llama Ajax, como se explica más adelante en la sección “Helpers de Ajax”.

Aquí casi no se incluye código Javascript, la razón es que la programación de scripts del lado del cliente de forma diferente: encapsula y abstrae toda la lógica Javascript en *helpers*, por lo que las plantillas no incluyen código Javascript.

Para el programador, añadir cierta lógica a un elemento de la página solo requiere de una línea de código PHP, pero la llamada a este *helper* produce código Javascript, cuya complejidad se puede comprobar al ver el código fuente de la página generada como respuesta. Los *helpers* se encargan de resolver los problemas de incompatibilidades entre navegadores por lo que la cantidad de código Javascript que generan puede ser muy importante.

Por tanto se ve como realizar los efectos que normalmente se programan manualmente con Javascript sin necesidad de utilizar Javascript. Todos los *helpers* se encuentran disponibles en las plantillas siempre que se declare de forma explícita el uso del *helper* llamado Javascript.

Algunos de estos *helpers* generan código HTML y otros generan directamente código Javascript. [symfony\_guia\_definitiva, 2007]

### 2.4.1.1 Javascript en las plantillas

En XHTML, los bloques de código Javascript deben encerrarse en secciones CDATA. Por eso es tedioso crear páginas que tienen muchos bloques de código Javascript. Por lo que se incluye un *helper* llamado `javascript_tag ()` y que transforma una cadena de texto en una etiqueta `<script>` válida según los estándares XHTML.

```
<? php echo javascript_tag("function mifuncion(){...}") ?>
=> <script type="text/javascript">
//<![CDATA[function mifuncion () {...}]]>
</script>
```

El uso habitual de Javascript, más que sus bloques de código, es la definición de enlaces que ejecutan un determinado script cuando se pincha en ellos.

```
<? php echo link_to_function('¡Pínchame!', "alert('Me has pinchado')") ?>
=> <a href="#" onClick="alert ('Me has pinchado'); return none ;"> ¡Pínchame! </a>
```

De la misma forma que el *helper* `link_to()` tiene una función relacionada llamada `button_to()`, también es posible ejecutar un script al pulsar un botón (`<input type="button">`) utilizando el *helper* `button_to_function()`. Si se necesita una imagen pinchable, se puede llamar a `link_to_function(image_tag('mi_imagen'), "alert('Me has pinchado')")`.  
[symfony\_guia\_definitiva, 2007]

### 2.4.1.2 Actualizando un elemento DOM

Una de las tareas habituales de las interfaces dinámicas es la actualización de algunos elementos de la página. Normalmente se realiza como se muestra:

```
<div id="indicador">Comienza el procesamiento de datos</div>
<?php echo javascript_tag("document.getElementById("indicador").innerHTML =
"<strong>El procesamiento de datos ha concluido</strong>");" ?>
```

También se incluye un *helper* que realiza esta tarea y que genera código JavaScript (no HTML). El *helper* se denomina `update_element_function ()` que actualiza un elemento mediante Javascript con dicho *helper*:

```
<div id="indicador">Comienza el procesamiento de datos</div>
<?php echo javascript_tag(update_element_function('indicador',
array('content' => "<strong>El procesamiento de datos ha concluido</strong>",)))?>
```

A primera vista parece que este *helper* no es muy útil, ya que el código necesario es tan largo como el código JavaScript original. En realidad su ventaja es la facilidad de lectura del código. Si lo que se necesita es insertar el contenido antes o después de un elemento, eliminarlo en vez de actualizarlo o no hacer nada si no se cumple una condición, el código JavaScript resultante es muy complicado. Sin embargo, el *helper* `update_element_function ()` permite mantener la facilidad de lectura del código de la plantilla, aquí se muestran opciones de dicho *helper*:

```
// Insertar el contenido después del elemento 'indicador'
update_element_function ('indicador', array('position' => 'after','content' =>
"<strong>El procesamiento de datos ha concluido</strong>"));
```

```
// Eliminar el elemento anterior a 'indicador', solo si $condicion vale true
update_element_function('indicador', array(
'action' => $condicion ? 'remove' : 'empty','position' => 'before',))
```

El *helper* permite que el código de las plantillas sea más fácil de entender que el código JavaScript, además de proporcionar una sintaxis unificada para efectos similares. También esa es la razón por la que el nombre del *helper* es tan largo: su nombre es tan explícito que no hace falta añadir comentarios que lo expliquen. [symfony\_guia\_definitiva, 2007]

### 2.4.2 Helpers de Ajax

¿Qué sucede si se quiere actualizar un elemento de la página no con JavaScript como se había dicho sino mediante un script de PHP que se encuentra en el servidor?

De esta forma, sería posible modificar parte de la página en función de una respuesta del servidor. El *helper* `remote_function()` realiza exactamente esa tarea, como se demuestra:

```
<div id="mizona"></div>
<?php echo javascript_tag(remote_function(array('update' => 'mizona', 'url' =>
'mimodulo/miaccion', )))?>
```

El parámetro `url` puede contener una URI interna (`modulo/accion?clave1=valor1&...`) o el nombre de una regla del sistema de enrutamiento, al igual que sucede con el *helper* `url_for()`. Cuando se ejecuta, el script anterior actualiza el contenido del elemento cuyo `id` es igual a `mizona` con la respuesta de la acción `mimodulo/miaccion`. Este tipo de interacción se llama Ajax, y es el núcleo de las aplicaciones web más interactivas.

Ajax permite que las páginas web respondan de forma más rápida mediante el intercambio en segundo plano de pequeñas cantidades de datos con el servidor, por lo que no es necesario recargar la página entera cada vez que el usuario realiza un cambio. El objetivo es aumentar la interactividad, la rapidez y la usabilidad de la página.

Ajax depende de `XMLHttpRequest`, un objeto Javascript cuyo comportamiento es similar a un *frame* oculto, cuyo contenido se puede actualizar realizando una petición al servidor y se puede utilizar para manipular el resto de la página web. Se trata de un objeto a muy bajo nivel, por lo que los navegadores lo tratan de forma diferente y el resultado es que se necesitan muchas líneas de código para realizar peticiones Ajax a mano. Afortunadamente, se usó la librería `Prototype` que encapsula todo el código necesario para trabajar con Ajax y proporciona un objeto Ajax mucho más simple además que se carga automáticamente cuando se utiliza un *helper* de Ajax en la plantilla.

Los helpers de Ajax no funcionan si la URL de la acción remota no pertenece al mismo dominio que la página web que la llama. Se trata de una restricción por motivos de seguridad que imponen los navegadores y que no puede saltarse.

Las interacciones de Ajax están formadas por 3 partes: el elemento que la invoca (un enlace, un formulario, un botón, un contador de tiempo o cualquier otro elemento que el usuario manipula e invoca la acción), la acción del servidor y una zona de la página en la que mostrar la respuesta de la acción.

Se pueden crear interacciones más complejas si por ejemplo la acción remota devuelve datos que se procesan en una función Javascript en el navegador del cliente. Symfony incluye numerosos helpers para insertar interacciones Ajax en las plantillas y todos contienen la palabra *remote* en su nombre. Además, todos comparten la misma sintaxis, un *array* asociativo con todos los parámetros de Ajax. Debe tenerse en cuenta que los helpers de Ajax generan código HTML, no código Javascript. [symfony\_guia\_definitiva, 2007]

### ¿Qué sucede con las acciones para Ajax?

Las acciones que se invocan de forma remota no dejan de ser acciones normales y corrientes. Se les aplica el sistema de enrutamiento, determinan la vista que deben generar en función del valor que devuelven, pasan variables a sus plantillas y pueden modificar el modelo como cualquier otra acción.

Sin embargo, cuando se invocan mediante Ajax, las acciones devuelven el valor true a la siguiente función:

```
$esAjax = $this->getRequest ()->isXmlHttpRequest();
```

Symfony es capaz de darse cuenta de que una acción se está ejecutando en un contexto Ajax y puede adaptar la respuesta de forma adecuada. Por tanto, y por defecto, las acciones Ajax no incluyen la barra de depuración de aplicaciones ni siquiera en el entorno de desarrollo. Además, no aplican el proceso de decoración (es decir, sus plantillas no se insertan por defecto en el layout correspondiente). Si se necesita decorar la vista de una acción Ajax, se debe indicar explícitamente la opción `has_layout: true` para su vista en el archivo `view.yml`.

Como el tiempo de respuesta es crucial en las interacciones Ajax, si la respuesta es sencilla, es una buena idea no crear la vista completa y devolver la respuesta directamente en forma de texto. Se puede utilizar por tanto el método `renderText ()` en la acción para no utilizar la plantilla y mejorar el tiempo de respuesta de las peticiones Ajax.

**Novedad introducida en la versión en desarrollo:** la mayoría de acciones Ajax finalizan con una plantilla que simplemente incluye un elemento parcial, porque el código de la respuesta Ajax ya se ha utilizado para mostrar la página inicial. Para evitar tener que crear una plantilla solo para una línea de código, la acción puede utilizar el método `renderPartial ()`. Este método se aprovecha de las ventajas de la reutilización de los elementos parciales, sus posibilidades de cache y la velocidad de ejecución del método `renderText ()`.

```
public function executeMiAccion ()
{
    // Código PHP de la acción
    return $this->renderPartial('mimodulo/miparcial');
}
```

### 2.4.3 Enlaces Ajax

Los enlaces Ajax constituyen una de las partes más importantes de las interacciones Ajax realizadas en las aplicaciones venideras. El helper `link_to_remote ()` muestra un enlace que llama a una función remota. La sintaxis es muy similar a `link_to ()`, excepto que el segundo parámetro es el array asociativo con las opciones Ajax, como muestra:

```
<div id="respuesta"></div>
<?php echo link_to_remote('Borrar este post', array('update' =>'respuesta','url' =>
'post/borrar?id='.$post->getId(),)) ?>
```

En el ejemplo anterior, al pulsar sobre el enlace “Borrar este post” se realiza una llamada en segundo plano a la acción `post/borrar`. La respuesta devuelta por el servidor se muestra automáticamente en el elemento de la página cuyo atributo `id` sea igual a `respuesta`. Aquí se ve la ejecución de una actualización remota mediante un enlace

También es posible utilizar una imagen en vez de texto para mostrar el enlace, utilizar el nombre de una regla de enrutamiento en vez de `modulo/accion` y añadir opciones a la etiqueta `<a>` como tercer argumento, tal y como se muestra:

```
<div id="emails"></div>
```

```
<?php echo link_to_remote(image_tag('refresh'), array('update' =>'emails','url' =>
'@listado_emails',), array('class' => 'enlace_ajax',)) ?> .[symfony_guia_definitiva,
2007]
```

### 2.4.4 Formularios Ajax

Los formularios web normalmente realizan una llamada a una acción que provoca que se deba recargar la página completa. El helper equivalente a `link_to_function()` para un formulario sería un helper que enviara los datos del formulario al servidor y que actualizara un elemento de la página con la respuesta del servidor. El helper `form_remote_tag()` crea una etiqueta `<form>` de apertura, como sucede con el helper `form_tag()`. El envío del formulario consiste en el envío en segundo plano de una petición de tipo POST a la acción `elemento/anadir` y con la variable `elemento` como parámetro de la petición.

La respuesta del servidor reemplaza los contenidos del elemento cuyo atributo `id` sea igual a `lista_elementos`, como se muestra en la figura. Los formularios Ajax se cierran con una etiqueta `</form>` de cierre de formularios. Aquí se ve como se ejecuta una actualización remota mediante un formulario.

Si es necesario incluir un formulario que sea normal y Ajax a la vez, lo mejor es definirlo como formulario normal y añadir, además del botón de envío tradicional, un segundo botón (`<input type="button" />`) para enviar el formulario mediante Ajax. Symfony define este botón mediante el helper `submit_to_remote()`. De esta forma, es posible definir interacciones Ajax que se degradan correctamente en los navegadores que no las soportan.

Los formularios más modernos no solo se encargan de enviar sus datos cuando el usuario pulsa sobre el botón de envío, sino que también pueden reaccionar a los cambios producidos por el usuario sobre alguno de sus campos. Symfony proporciona el helper `observe_field()` para realizar esa tarea. Este helper crea un sistema que sugiere valores a medida que el usuario escribe sobre un campo: cada carácter escrito en el campo elemento lanza una petición Ajax que actualiza el valor del elemento `sugerencias_elemento` de la página.

El par *modulo/accion* correspondiente a la regla *@elemento\_escrito* se ejecuta cada vez que el usuario modifica el valor del campo de formulario que se está observando (en este caso, elemento) sin necesidad de enviar el formulario. La acción puede acceder a los caracteres escritos en cada momento por el usuario mediante el parámetro *elemento* de la petición.

Si se necesita enviar otro valor en vez del contenido del campo de formulario que se está observando, se puede especificar en forma de expresión JavaScript en el parámetro *with*. Si por ejemplo es necesario que la acción disponga de un parámetro llamado *param*, se puede utilizar el helper **observe\_field** (). Si se quieren observar todos los campos de un formulario, se puede utilizar el helper **observe\_form** (), que llama a una función remota cada vez que se modifica uno de los campos del formulario.

### 2.4.5 Creando efectos visuales

Symfony integra los efectos visuales y para poder incluir estos efectos más avanzados que simplemente mostrar y ocultar elementos `<div>` en las páginas. Se proporcionan objetos y funciones JavaScript que manipulan el DOM de la página para conseguir efectos visuales complejos. Aquí se encapsula el objeto *Effect* de JavaScript en un helper llamado **visual\_effect** (), que forma parte del helper Javascript. El código generado es JavaScript y puede utilizarse en un enlace normal.

El elemento indicador aparece de forma progresiva cuando comienza la petición Ajax y se desaparece también progresivamente cuando se recibe la respuesta del servidor. Además, el elemento respuesta se resalta después de ser actualizado por la petición remota, de forma que esa parte de la página capte la atención del usuario.

### 2.5 Estrategia para la captura de errores

Como todo sistema, para garantizar su correcto funcionamiento se debe tener en cuenta un tratamiento de errores, hacer captura de las excepciones que son lanzadas por el sistema para advertir al usuario de que la cosa va mal y debe ser cambiada, el sistema captura todas aquellas excepciones que son lanzadas y se le da un tratamiento para que el sistema no colapse.

Se sabe de antemano que a la hora de registrar un nuevo usuario por poner un ejemplo, al introducir el número de CI o algún otro dato que este incorrecto nuestro sistema debe ser capaz de advertir al usuario en caso contrario, esta excepción es lanzada por el sistema.

De esta manera se da solución a las problemáticas de los lanzamientos de excepciones que se dan en el sistema según las peticiones del usuario, utilizando los try - catch, funciones que posibilitan hacer capturas de errores, capturando los tipos de errores lanzados y mostrando de manera visible al usuario mensajes de confirmación, el cual le dará una idea de que anda mal y como solucionarlo.

### **2.6 Conclusiones del capítulo**

En este capítulo se dio una panorámica de cómo se da respuesta al problema, la solución propuesta, para ello se abordó temas, sin los cuales no hubiese sido posible la realización del sistema. Como detalle principal se habló de la capa de Presentación la cual brinda interfaces que le permiten al usuario intercambiar con el sistema.

Se dio una breve descripción de las interfaces de usuario que contiene la capa de Presentación, cumpliendo los requisitos de forma correcta en ambos módulos.

También trató sobre las estrategias de integración con AJAX que satisfacen nuestro sistema y la importancia de las mismas para mayor organización y rapidez de la aplicación, así como la estrategia para la captura de errores en los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

# 3

## Capítulo 3: Validación de la solución propuesta

### 3.1 Introducción al capítulo

En el presente capítulo se exponen los resultados de las pruebas de validación de interfaces realizadas a cada uno de los CU implementados por el programador en la capa de Presentación de los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.

### 3.2 Descripción de las pruebas de Validación de Interfaces de Usuario

Las pruebas de validación de las interfaces de usuario básicamente chequean para ver si los tipos y rangos de los valores de un campo son correctos. Por ejemplo, si un campo numérico es realmente un número o si una fecha es realmente una fecha válida, etc. La validación de datos es cuando se necesita chequear si el valor dado es válido entre, dígame, una lista de posibles elecciones, quizás requiriendo una búsqueda en una base de datos. [Juristo - Moreno –Vegas, 2006]

A continuación se muestran las pruebas de validación de interfaces, para cada una de las interfaces de usuario de los módulos “Selección al Ingreso” y “Estructura y Composición”, que certifican la calidad de estas interfaces:

#### 3.2.1 Modulo “Selección al Ingreso”

➤ *CU Establecer Necesidades de Convocatoria:*

<b>Caso de uso:</b> CU Establecer Necesidades de Convocatoria
<b>Caso de prueba:</b> Verificar el establecimiento de la necesidad de convocatorias.

<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se escoge la opción ***** del menú.</li> <li>• Se selecciona una plaza vacante.</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se le asigna a la plaza una ficha.</li> <li>• Marca la ficha a la que se le asigna la plaza registrándola en el cargo final de la misma.</li> <li>• Se le cambia el estado a la plaza por reservada.</li> <li>• Se cambia el estado de la ficha por pendiente a Aceptación.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como psicólogo.</li> <li>• Que existan plazas vacantes.</li> <li>• Que haya fichas en el estado de Bolsa (“0000011111” ó “0000011101”) para dicha plaza.</li> </ul>
<p><b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García</p>

➤ *CUVerificarPlazasVacantes:*

<p><b>Caso de uso:</b> CUVerificarPlazasVacantes</p>
<p><b>Caso de prueba:</b> Verificar las plazas vacantes.</p>
<p><b>Entrada:</b></p> <p>Que sea llamado el caso de uso Culminar Ficha o Establecer Necesidades de</p> <ul style="list-style-type: none"> <li>• Convocatoria</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Busca todas las plazas que se encuentren vacantes en una unidad seleccionada</li> </ul>

anteriormente.

**Condiciones:**

- Que se haya ejecutado el caso de uso Culminar Ficha.
- Que se haya ejecutado el caso de uso Establecer Necesidades de Convocatoria.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ **CUCrearFichas:**

**Caso de uso:** CUCrearFichas

**Caso de prueba:** Verificar la creación de una ficha.

**Entrada:**

- Se introduce el Carnet de Identidad del candidato en un TextField.
- Se seleccionan los paneles para obtener las opciones del candidato.
- Se seleccionan las opciones para ese candidato, en cada uno de los paneles.

**Resultado:**

Se presiona el botón Guardar, se limpia el formulario, además se actualizan todos los datos en la BD y se regresa al menú para dar la posibilidad de introducir un nuevo carnet de Identidad y crear una nueva ficha.

En caso de que el Carnet del candidato coincida con alguno existente en la Base de Datos y haya sido desertor, se pregunta si se desea continuar el proceso, en ese caso se cargan los datos relativos a ese candidato existentes en la BD; en caso contrario no realiza ninguna acción. Por otra parte si no ha sido desertor se levantan los datos existentes a dicho candidato.

**Condiciones:**

- Que el actor este autenticado como psicólogo.
- Las opciones deben estar registradas en la BD.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CUResultadosPrimeraEntrevista:***

<b>Caso de uso:</b> CUResultadosPrimeraEntrevista
<b>Caso de prueba:</b> Verificar los resultados de la primera entrevista.
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se selecciona la opción Resultados Primera Entrevista en el menú.</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se cargan todas las fichas que se encuentran en el estado de Creada.</li> <li>• Se registra alguna observación que se tomo en la primera entrevista de ser necesario</li> <li>• Se registra si se le continuara el proceso al candidato o no.</li> <li>• En caso de que el cargo necesite psicométrico se le cambia el estado a la ficha por Presentada (“0000000001”).</li> <li>• Si el cargo no necesita psicométrico se le cambia el estado a la ficha por Presentada-Pendiente a Verificación (“0000000101”).</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como psicólogo.</li> <li>• Que la ficha este en estado de Creada (“0000000000 binario”).</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ *CUResultadosAnálisisPsicologico:*

<b>Caso de uso:</b> CUResultadosAnálisisPsicologico
<b>Caso de prueba:</b> Verificar los resultados del análisis al Psicológico.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>Se escoge la opción Análisis Psicológico del menú.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>Guarda la observación pública, codifica y guarda la observación.</li> <li>Se cambia el estado a la ficha por Pendiente a Verificación (“0000000111”).</li> </ul>
<b>Condiciones:</b> <ul style="list-style-type: none"> <li>Que el actor este autenticado como psicólogo.</li> <li>Que la ficha se encuentre en estado de Presentada (0000000001).</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ *CUResultadosVerificacion:*

<b>Caso de uso:</b> CUResultadosVerificacion
<b>Caso de prueba:</b> Verificar los resultados de la verificación.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>Se escoge la opción Resultados Verificación del menú.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>Se cambia el estado a la ficha por Verificada en caso de que venga directo del estado de Presentada (0000001101) y en caso de que venga del estado de Psicométrico (0000001111).</li> </ul>
<b>Condiciones:</b>

- Que el actor este autenticado como psicólogo.
- Que la ficha se encuentre en estado Pendientes a Verificación donde los binarios pueden ser (“0000000111” ó “0000000101”) indistintamente si se les hace el psicométrico o no.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CUPendientesaAceptacion:***

**Caso de uso:** CUPendientesaAceptacion

**Caso de prueba:** Verificar la aceptación de los candidatos pendientes.

**Entrada:**

- Se escoge la opción Pendiente a Aceptación del menú.

**Resultado:**

- Se confirma la aceptación de la plaza o marca al candidato como desertor.

**Condiciones:**

- Que el actor este autenticado como psicólogo.
- Que existan fichas que se encuentran en el estado de Pendientes a Aceptación.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CUPlanificarDiasPsicometrico:***

**Caso de uso:** CUPlanificarDiasPsicometrico

**Caso de prueba:** Verificar la planificación de los días del psicométrico.

**Entrada:**

- Se escoge la opción Planificar Días Psicométrico del menú.

<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se crean los días para realizar el psicométrico además del número de personas y el local donde se realizará.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como psicólogo.</li> <li>• Que existan fichas que se encuentran en el estado de Pendientes a Aceptación.</li> </ul>
<p><b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García</p>

➤ **CUPlanificarPsicometrico:**

<p><b>Caso de uso:</b> CUPlanificarPsicometrico</p>
<p><b>Caso de prueba:</b> Verificar la planificación del psicométrico.</p>
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se escoge la opción Planificar fichas para la Prueba del menú.</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se planifica que candidatos van a asistir a un determinado día de psicométrico.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como psicólogo.</li> <li>• Que el caso de uso Planificar días Psicométrico se haya ejecutado satisfactoriamente</li> <li>• Que existan fichas con el estado Presentada (“0000000001”).</li> </ul>
<p><b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García</p>

➤ ***CUCancelarPsicometrico:***

<b>Caso de uso:</b> CUCancelarPsicometrico
<b>Caso de prueba:</b> Verificar la cancelación de un día de psicométrico.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>Se escoge la opción Cancelar Psicométrico del menú.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>Se cancela el día del Psicométrico seleccionado.</li> </ul>
<b>Condiciones:</b> <ul style="list-style-type: none"> <li>Que el actor este autenticado como psicólogo.</li> <li>Que exista el día que se quiere cancelar previamente planificado.</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CURegistrarAsistenciaalPsicometrico:***

<b>Caso de uso:</b> CURegistrarAsistenciaalPsicometrico
<b>Caso de prueba:</b> Verificar la asistencia al psicométrico.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>Se escoge la opción Registrar Asistencia Psicométrico del menú.</li> <li>Se selecciona un día de los planificados para realizar psicométrico.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>Se marca las fichas a las cuales se les realizó la prueba.</li> <li>Se le cambia el estado a todas las fichas marcadas por el estado Psicométrico ("0000000011").</li> </ul>
<b>Condiciones:</b>

- Que el actor este autenticado como psicólogo.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ **CUCrearAgenda:**

**Caso de uso:** CUCrearAgenda

**Caso de prueba:** Verificar la creación de una agenda.

**Entrada:**

- Se escoge la opción crear agenda del menú.  
Se selecciona entre los candidatos los que serán llevados a la reunión de órganos colegiados ya sea para la aprobación de la contratación o de trabajadores para la creación de un anexo, o de un nuevo contrato según las fichas.

**Resultado:**

- Adiciona la persona para la planificación de la agenda.
- Se crea dicha Agenda.
- El sistema imprime dicha agenda.

**Condiciones:**

- Que el usuario este autenticado como Secretario de Órganos Colegiados.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ **CUAprobacionCandidatos:**

**Caso de uso:** CUAprobacionCandidatos

**Caso de prueba:** Verificar la aprobación de candidatos.

**Entrada:**

- Se selecciona la opción Comisión de Órganos Colegiados en el menú.

**Resultado:**

- Se Selecciona una persona y la aprueba.
- Se escribe el número del acuerdo de aprobación de dicha persona.
- Le cambia el estado en caso de que sea un candidato a la ficha por Aceptado Órganos Colegiados.
- En caso de que sea un trabajador y este pendiente a un nuevo contrato se le cambia el estado a Pendiente a Renovar Contrato
- En caso de que el trabajador se encuentre pendiente a anexos se le cambia el estado a Pendiente a Anexos.
- Le cambia el estado en la agenda a Tratado.
- Para la comisión de Cuadros se genera la Resolución del Nombramiento.
- Para la comisión Representativa se genera el Acuerdo de Nombramiento

**Condiciones:**

- Que se haya ejecutado el caso de uso Crear Agenda.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ *CUCulminarFicha:*

**Caso de uso:** CUCulminarFicha

**Caso de prueba:** Verificar la correcta conclusión del proceso de creación de una ficha.

**Entrada:**

- Se escoge la opción Terminar Proceso del menú.
- Se cargan todas las fichas que se encuentran en el estado Verificación.

<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se le asigna la plaza al candidato según las plazas vacantes.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como psicólogo.</li> <li>• Que existan plazas vacantes.</li> <li>• Que la ficha se encuentre en el estado de Verificación (“0000001111” ó “0000001101”).</li> </ul>
<p><b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García</p>

➤ **CUAsignarPlaza:**

<p><b>Caso de uso:</b> CUAsignarPlaza</p>
<p><b>Caso de prueba:</b> Verificar la asignación de una plaza.</p>
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Este caso de uso se inicia por el sistema en el paso 10 del caso de uso Culminar Ficha o en el paso 3 del caso de uso Pendientes a Aceptación.</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se le asigna la plaza al candidato según las plazas vacantes.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como psicólogo.</li> <li>• Que existan plazas vacantes.</li> <li>• Que la ficha se encuentre en el estado de Verificación (“0000001111” ó “0000001101”).</li> </ul>
<p><b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García</p>

➤ **CUMarcarelCandidatocomoDesertor:**

<b>Caso de uso:</b> CUMarcarelCandidatocomoDesertor
<b>Caso de prueba:</b> Verificar el marcado de un candidato como desertor.
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Que sea llamado el caso de uso del proceso.</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Busca la última ficha por el número de carnet de identidad del candidato.</li> </ul> <p>Le cambia el estado a Desertor (“001xxxxxx” donde las “x” pueden ser 0 ó 1 indistintamente,</p> <ul style="list-style-type: none"> <li>• esto es porque el candidato puede ser marcado como desertor en cualquier momento del proceso).</li> <li>• Se guarda la observación del motivo de la deserción.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que exista la ficha de dicho candidato.</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ **CUAprobacionCandidatos:**

<b>Caso de uso:</b> CUAprobacionCandidatos
<b>Caso de prueba:</b> Verificar la aprobación de candidatos.
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se selecciona la opción Comisión de Órganos Colegiados en el menú.</li> </ul>

<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se Selecciona una persona y la aprueba.</li> <li>• Se escribe el número del acuerdo de aprobación de dicha persona.</li> <li>• Le cambia el estado en caso de que sea un candidato a la ficha por Aceptado Órganos Colegiados.</li> <li>• En caso de que sea un trabajador y este pendiente a un nuevo contrato se le cambia el estado a Pendiente a Renovar Contrato</li> <li>• En caso de que el trabajador se encuentre pendiente a anexos se le cambia el estado a Pendiente a Anexos.</li> <li>• Le cambia el estado en la agenda a Tratado.</li> <li>• Para la comisión de Cuadros se genera la Resolución del Nombramiento.</li> <li>• Para la comisión Representativa se genera el Acuerdo de Nombramiento</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que se haya ejecutado el caso de uso Crear Agenda.</li> </ul>
<p><b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García</p>

### 3.2.2 Modulo “Estructura y Composición”

➤ **CUCrearUnidad:**

<p><b>Caso de uso:</b> CUCrearUnidad</p>
<p><b>Caso de prueba:</b> Verificar la creación de una unidad.</p>
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se selecciona la opción Archivo Crear Unidad en el menú.</li> </ul>
<p><b>Resultado:</b></p>

- Se seleccionan las opciones para dicha unidad.
- Se crea la unidad.

**Condiciones:**

- Que el actor este autenticado como Jefe de Recursos Humanos o Técnico de Recursos Humanos.
- Que exista una resolución del Jefe de la Aduana General de la República.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CUCrearSubUnidad:***

**Caso de uso:** CUSCrearSubUnidad

**Caso de prueba:** Verificar la creación de una sub unidad.

**Entrada:**

- Se selecciona la opción Archivo Crear Sub Unidad en el menú.

**Resultado:**

Registra los datos de la Sub Unidad.

- Se crea la Sub Unidad.
- Se subordina la sub-unidad a la unidad seleccionada.
- 

**Condiciones:**

- Debe existir al menos una Unidad creada.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CUCrearUnidadOrganizativa:***

<b>Caso de uso:</b> CUCrearUnidadOrganizativa
<b>Caso de prueba:</b> Verificar la creación de una unidad organizativa.
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se selecciona la opción Archivo Crear Unidad Organizativa en el menú.</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se seleccionan las opciones para dicha unidad organizativa.</li> <li>• Se crea la unidad.</li> <li>• Se subordina la unidad a una unidad o sub unidad seleccionada.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como Jefe de Recursos Humanos.</li> <li>• Deben existir Unidades o Sub-Unidades creadas.</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CUCrearGrupo:***

<b>Caso de uso:</b> CUCrearGrupo
<b>Caso de prueba:</b> Verificar la creación de un grupo de trabajo.
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se selecciona la opción Archivo Crear Grupo Trabajo en el menú.</li> </ul>

<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se seleccionan las opciones para dicho grupo.</li> <li>• Se crea el grupo.</li> <li>• Se subordina el grupo a una unidad o sub unidad seleccionada.</li> </ul>
<p><b>Condiciones:</b></p> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como Jefe de Recursos Humanos.</li> <li>• Deben existir Unidades o Sub-Unidades creadas.</li> </ul>
<p><b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García</p>

➤ ***CUAsignarPatrimonio:***

<p><b>Caso de uso:</b> CUAsignarPatrimonio</p>
<p><b>Caso de prueba:</b> Verificar la asignación de patrimonio a una unidad.</p>
<p><b>Entrada:</b></p> <ul style="list-style-type: none"> <li>• Se selecciona una unidad de la estructura de la Aduana General de la República (AGR).</li> </ul>
<p><b>Resultado:</b></p> <ul style="list-style-type: none"> <li>• Se registra el centro de costo al que pertenece la unidad, especificando si es propio o no.</li> <li>• Se le asigna patrimonio a dicha unidad.</li> </ul>
<p><b>Condiciones:</b></p>

- Que el actor este autenticado como Jefe de Recursos Humanos.
- Deben existir unidades creadas.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CURealizarAmortizacion:***

<b>Caso de uso:</b> CURealizarAmortizacion
<b>Caso de prueba:</b> Verificar la amortización de una unidad.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>• Selecciona un elemento de la estructura y la opción Amortizar.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>• Se elimina la plaza.</li> <li>• Se genera la planilla de amortización de plazas.</li> </ul>
<b>Condiciones:</b> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como Jefe de Recursos Humanos.</li> <li>• Deben existir plazas subordinadas a algún Grupo de trabajo.</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CURealizarConversionesdePlazas:***

<b>Caso de uso:</b> CURealizarConversionesdePlazas
<b>Caso de prueba:</b> Verificar la conversión de una unidad.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>• Selecciona un elemento de la estructura y la opción Conversión de plaza.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>• Se selecciona el nuevo cargo de la plaza.</li> <li>• Se genera la planilla de Conversión de Plaza en la que aparece el cargo anterior y el cargo actual.</li> </ul>
<b>Condiciones:</b> <ul style="list-style-type: none"> <li>• Que el actor este autenticado como Jefe de Recursos Humanos.</li> <li>• Deben existir plazas creadas.</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ **CURealizarIncrementos:**

<b>Caso de uso:</b> CURealizarIncrementos
<b>Caso de prueba:</b> Verificar el incremento de nuevas plazas.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>• Selecciona un elemento de la estructura y la opción Incrementos de Plazas.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>• Se ejecuta el caso de uso Crear Plaza.</li> </ul>

- Se genera una planilla de Incremento de Plazas.

**Condiciones:**

- Que el actor este autenticado como Jefe de Recursos Humanos.
- Deben existir plazas creadas.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

➤ ***CUGestionarPlaza:***

**Caso de uso:** CUGestionarPlaza

**Caso de prueba:** Verificar la gestión de una plaza.

**Entrada:**

- Selecciona un elemento de la estructura.

**Resultado:**

- Se ejecutan indistintamente los Casos de uso: **Realizar Amortización, Realizar Conversiones de Plazas, Realizar Incrementos**, dado que este caso de uso es una unión de los anteriormente mencionados.
- Se generar la planilla de Conversiones de Plazas.

**Condiciones:**

- Que el actor este autenticado como Jefe de Recursos Humanos.
- Deben existir plazas subordinadas a algún Grupo de trabajo.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

### ➤ *CUAsignarPersonalidadJurídica:*

<b>Caso de uso:</b> CUAsignarPersonalidadJurídica
<b>Caso de prueba:</b> Verificar la asignación de una personalidad jurídica.
<b>Entrada:</b> <ul style="list-style-type: none"> <li>• Selecciona un elemento de la estructura y la opción asignar Personalidad Jurídica.</li> </ul>
<b>Resultado:</b> <ul style="list-style-type: none"> <li>• Se registra la información referente al número de la resolución del Ministerio de Economía y Planificación (MEP).</li> <li>• Se registra la fecha de vigencia de dicha resolución.</li> </ul>
<b>Condiciones:</b> <ul style="list-style-type: none"> <li>• Debe existir una resolución del Ministerio de Economía y Planificación (MEP).</li> <li>• Deben existir Unidades creadas.</li> </ul>
<b>Elaborado por:</b> Yiliexy Martínez García – Hugo Leonardo Benítez García

### ➤ *CUEditarEstructura:*

<b>Caso de uso:</b> CUEditarEstructura
<b>Caso de prueba:</b> Verificar la asignación de una personalidad jurídica.
<b>Entrada:</b> <p>Este caso de uso se puede realizar en dos momentos.</p> <ul style="list-style-type: none"> <li>• <ol style="list-style-type: none"> <li>1. Para crear nuevos componentes de una estructura subordinadas a una unidad que no posee categoría.</li> <li>2. Cuando se crea una unidad que posee categoría y se realizan modificaciones a la</li> </ol> </li> </ul>

estructura propuesta por el sistema

**Resultado:**

- Ejecuta el caso de uso **Crear Plaza, Crear Grupo, Crear Sub-Unidad o Crear Unidad Organizativa.**

**Condiciones:**

- Que el actor este autenticado en el sistema como Jefe de Recursos Humanos
- Que exista una unidad creada para aplicar cualquiera de las operaciones relacionadas.

**Elaborado por:** Yiliexy Martínez García – Hugo Leonardo Benítez García

**3. 3 Conclusiones del capítulo**

En este capítulo se analizaron los resultados de las pruebas de validación de interfaz de usuario, realizadas a cada uno de los casos de uso que conforman los módulos “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos. Mediante estas pruebas se pueden certificar cada una de estas interfaces que, de forma amigable, funcional y rápida, se integran con la capa de Presentación y culmina todo un proceso de desarrollo de estos dos módulos del Sistema.

## **CONCLUSIONES GENERALES**

- Se realizó un estudio acerca de las interfaces gráficas de usuario
- Se abordaron los principios básicos de la programación, además de las tareas fundamentales que debe desarrollar el programador, según la metodología seleccionada para el desarrollo del sistema.
- Se realizó un estudio sobre las herramientas definidas para el desarrollo del Sistema.
- Teniendo en cuenta la librería ExtJs se expone la estructura de la capa de Presentación para los módulos: “Selección al Ingreso” y “Estructura y Composición” del Sistema Integral para la Gestión de Recursos Humanos.
- Se implementaron y validaron las interfaces de usuario, de los módulos: “Selección al Ingreso” y “Estructura y Composición”, integradas con la capa de Presentación desarrollada para el sistema.
- Se expone el resultado del trabajo del programador, acorde con la planificación, para el cumplimiento de la implementación y prueba de cada interfaz de usuario.

## **RECOMENDACIONES**

- La implementación de la capa de Presentación, para los módulos restantes del Sistema Integral para la Gestión de Recursos Humanos, teniendo en cuenta el estándar de codificación y la estructura de la capa de Presentación definida.
  
- Crear una Ayuda y un Manual del usuario.

**REFERENCIAS BIBLIOGRAFICAS**

- [Abelson - Sussman, 1996] Abelson, H. y. S., G.J. con Sussman, J. (1996). "Structure and Interpretation of Computer Programs". Url: <http://mitpress.mit.edu/sicp/full-text/book/book.html>.
- [Bonanata, 2003] Bonanata, M. (2003). "Programación y Algoritmos".
- [Booch, 1996] Booch, G. (1996). "Análisis y Diseño orientado a Objetos con Aplicaciones". Url: <http://catalogo.info.unlp.edu.ar/cgi-bin/koha/opac-detail.pl?bib=1427>.
- [Bush, 1945] Bush, V. (1945). "As we may think", Url: <http://web.mit.edu/STS.035/www/PDFs/think.pdf>.
- [Collado, 2005] Collado, M. (2005). "Pruebas de Software". Url: <http://lml.ls.fi.upm.es/~mcollado/ed2/pruebas.ppt>.
- [Figuroa, 2005] Figuroa, Z. J. H. (2005). "Fundamentos de estructuras de datos: soluciones en Ada, Java y C++".  
Url: <http://books.google.com/books?id=KSE4HwF8BmcC&pg=PA458&ots=n49WibUYhd&dq=Fundamentos+de+la+programaci%C3%B3n+Algoritmos+y+estructuras+de+datos&sig=vUhZ093YzDxUti-2EWe42DInZEI>
- [Floyd, 1979] Floyd. R. (1979). "The Paradigms of Programming".
- [Fuente, 2000] Fuente, J. S.-C. d. I. (2000). "Thomas Samuel Kuhn, Biografía". Url: <http://www.webdianoia.com/contemporanea/kuhn.htm>.
- [Pinson - Wiener, 2000] Lewis J. Pinson, R. S. W. (2000). "Fundamentals of OOP and Data Structures in Java", Url: [http://books.google.com/books?id=m0vpxjdjUJYC&pg=PA3&ots=\\_n-NvsUMvS&dq=Fundamentals+of+Data+Structures&sig=2KdHbXgaDtN0sdPMtwB6kTfLcd4](http://books.google.com/books?id=m0vpxjdjUJYC&pg=PA3&ots=_n-NvsUMvS&dq=Fundamentals+of+Data+Structures&sig=2KdHbXgaDtN0sdPMtwB6kTfLcd4)
- [Mañas, 2007] Mañas, J. A. (1999). "Prueba de Programas".  
Url: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.
- [Horta-Ostos-Gutiérrez, 1991] Patricia Horta, M. O. y. O. G. (1991). "Diseño estructurado de algoritmos". Url: <http://www.itver.edu.mx/comunidad/material/algoritmos/>.
- [Reimer, 2005] Reimer, J. (2005). "A History of the GUI", Url: <http://arstechnica.com/articles/paedia/gui.ars>.
- [Lewis-Rieman, 1993] Rieman, C. L. y. J. (1993). "Task-centered user interface design".  
Url: <http://hcbib.org/tcuid/>.

[Bobrow-Stefik, 1986] Stefik, D. G. B. y. M. J. (1986). "Perspectives on Artificial Intelligence Programming", 23 Url: <http://www2.parc.com/istl/groups/hdi/papers/stefikperspectives-on-AI-programming.pdf>

[Stephanidis, 2001] Stephanidis, C. (2001). "User Interfaces for All". Url: <http://books.google.com/books?id=kGslhSbZqzkC&pg=PA3&ots=5VCbLMr-5H&dq=Interfaces&sig=sjh97iBXzUoXmkTU2hJ0IPaSCQA>

[Cormen-Leiserson-Rivest-Stein, 2001] Thomas H. Cormen, C. E. L., Ronald L. Rivest, y Cliff Stein. (2001). "Introduction to Algorithms", Url: [http://books.google.com/books?id=NLngYyWFI\\_YC&dq=Introduction+to+Algorithms&pg=PP1&ots=BvUvIE5nB8&sig=uW7hPRKgvlpBLmpLxF2jaqy0XY&prev=http://www.google.com/search%3Fhl%3Den%26q%3DIntroduction%2Bto%2BAlgorithms%26btnG%3DSearch&sa=X&oi=print&ct=title](http://books.google.com/books?id=NLngYyWFI_YC&dq=Introduction+to+Algorithms&pg=PP1&ots=BvUvIE5nB8&sig=uW7hPRKgvlpBLmpLxF2jaqy0XY&prev=http://www.google.com/search%3Fhl%3Den%26q%3DIntroduction%2Bto%2BAlgorithms%26btnG%3DSearch&sa=X&oi=print&ct=title)

[Juristo - Moreno –Vegas, 2006]

[symfony\_guia\_definitiva, 2007]

## **GLOSARIO DE TERMINOS**

**Hardware:** Conjunto de componentes materiales de un sistema informático. Cada una de las partes físicas que forman un ordenador, incluidos sus periféricos.

**Interactivo:** Un sistema es interactivo cuando permite un diálogo continuo entre el usuario y la aplicación, respondiendo ésta a las órdenes de aquel.

**Interfaz:** Zona de contacto o conexión entre dos componentes de "hardware"; entre dos aplicaciones; o entre un usuario y una aplicación. Apariencia externa de una aplicación informática.

**Librerías:** En Inglés library. Cuando se habla de ordenadores, se refiere al conjunto de rutinas que realizan las operaciones usualmente requeridas por los programas. Las librerías pueden ser compartidas, lo que quiere decir que las rutinas de la librería residen en un fichero distinto de los programas que las utilizan. Los programas enlazados con bibliotecas compartidas no funcionarán a menos que se instalen las bibliotecas o librerías necesarias.

**Multiplataforma:** Multiplataforma es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

**Sistema:** Un sistema es una colección de unidades organizadas para cumplir un propósito en específico. Un sistema puede ser descrito por uno o más modelos y posiblemente desde diferente.

**Usuario:** Es el que recibe los estímulos de parte del sistema y a su vez se encarga de retroalimentarlo y definir su comportamiento.

**Usabilidad:** El modelo conceptual de la usabilidad, proveniente del diseño centrado en el usuario, no está completo sin la idea utilidad.

## ANEXOS

### Diccionario de Datos

#### Modulo “Selección al Ingreso”

➤ *CUCrearFichas:*

#### Datos Personales

**Tabla 1 Datos Personales**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Número de Carnet de Identidad	Cadena de caracteres	No	Dígitos del 0 al 9, longitud de 11 caracteres		En este campo se registra el número de carnet de identidad del candidato.
Nombre	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		Este campo guarda el nombre del candidato que será procesado.
Segundo Nombre	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		Este campo guarda el segundo nombre del candidato que será procesado.
Primer Apellido	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		Este campo guarda el primer apellido del candidato que será procesado.
Segundo Apellido	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		Este campo guarda el segundo apellido del candidato que será procesado.
Foto	Imagen	Si			En este campo se debe cargar la foto del candidato, si la presenta.
Fecha de nacimiento	Fecha	No	Fecha en formato dd/mm/aaaa		Este campo guarda la fecha de nacimiento del candidato.
Municipio de nacimiento	Cadena de caracteres	Si	Todos los municipios de la provincia		Este campo guarda el municipio donde nació el candidato.

to			seleccionada.		
Provincia de nacimiento	Cadena de caracteres	Si	Pinar del Rio Ciudad de la Habana La Habana Matanzas Cienfuegos Villa Clara Sancti Spíritus Ciego de Avila Camagüey Las Tunas Holguín Granma Santiago de Cuba Guantánamo Isla de la Juventud		Este campo guarda la provincia donde nació el candidato.
Sexo	Carácter	No	M F		Este campo guarda el tipo de sexo al cual pertenece el candidato
Numero de Hijos	Número entero	No	Dígitos del 0 al 9		Este campo guarda el número de hijos que tiene el candidato.
Estado Civil	Cadena de caracteres	No	Soltero Casado Divorciado Viudo Unión Consensual.		Este campo guarda el estado civil del candidato.
Raza	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		Este campo guarda la raza a la que pertenece el individuo.

**Datos de Contacto**
**Tabla 2 datos de Contacto**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no válidas	
Tipo	Cadena de caracteres	No	Domicilio Ultimo Trabajo		En este campo se debe escoger el tipo contacto con el candidato.
Calle	Cadena de caracteres	No	Todas las letras del vocabulario en español,		En este campo se guarda el nombre de la calle de dicha dirección.

			además de las vocales acentuadas.		
Entre	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha dirección.
Y	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha dirección.
Edificio	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número o nombre de dicho edificio.
No	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número de la casa de dicha dirección.
Apto	Cadena de	Si	Todas las letras		En este campo se guarda el

	caracteres		del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		número del apartamento de dicha dirección.
Barrio	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el nombre o numero del barrio a donde pertenece la dirección.
Provincia	Cadena de caracteres	No	<p>Pinar del Rio</p> <p>Ciudad de la Habana</p> <p>La Habana</p> <p>Matanzas</p> <p>Cienfuegos</p> <p>Villa Clara</p> <p>Sancti Spíritus</p> <p>Ciego de Avila</p> <p>Camagüey</p> <p>Las Tunas</p> <p>Holguín</p> <p>Granma</p> <p>Santiago de Cuba</p> <p>Guantánamo</p> <p>Isla de la Juventud</p>		En este campo se guarda el nombre de la provincia de dicha dirección.

Municipio	Cadena de caracteres	No	Todos los municipios correspondientes a la provincia seleccionada		En este campo se guarda el nombre del municipio de dicha dirección.
Teléfono	Cadena de caracteres	Si	Los dígitos del 0 al 9 Carácter -		En este campo se guarda el número de teléfono de dicha dirección.

**Datos Formación**
**Tabla 2 Datos Formación**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Escolaridad Vencida	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda la última escolaridad vencida de la cual tengo titulación.
En el Centro de Enseñanza	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el nombre del centro de enseñanza donde se expidió dicho título.
Del Municipio	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el municipio donde se expidió dicho título.
De la Provincia	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la provincia donde se expidió dicho título.
Tomo	Cadena de	No	Todas las letras		En este campo se guarda el

	caracteres		del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		tomo de dicho título.
Folio	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el folio de dicho título.
Fecha	Fecha	No	Fecha en formato dd/mm/aaaa		En este campo se guarda la fecha de cuando se emitió este título.
Nombre	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el nombre de la persona que fue el encargado de emitir el título.
Apellido	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el apellido de la persona que fue el encargado de emitir el título.
Cargo	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el cargo de la persona que fue la encargada de emitir el título.

### Complementarios

**Tabla 3 Otros datos**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Cargo	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales		En este campo se guarda el cargo por el que se le esta iniciando el proceso al candidato.

			acentuadas.		
Forma de presentación	Radiobutton	No	Espontanea Convocatoria Recomendación		En dicho campo se guarda la forma de presentación del candidato.
Nombre sanción	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el nombre de la sanción que se le aplico al candidato.
Clasificación de la sanción	Cadena de caracteres	Si	Laboral Tribuna Políticamente		En este campo se guarda la clasificación de dicha sanción.
Fecha	Fecha	Si	Fecha en formato dd/mm/aaaa		En este campo se guarda la fecha en la que se aplico la sanción.
Motivo de la sanciones	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda la descripción del porque se le aplico la sanción al candidato.
Fecha	Fecha	Si	Fecha en formato dd/mm/aaaa		
País	Cadena de caracteres	Si	Todos los países existentes en una tabla de la base de datos.		En este campo se escoge el país a donde realizo dichos trámites migratorios.
Motivos	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda los motivos por los que el candidato realizo dichos trámites.
Militancia	Radiobutton	Si	PCC UJC Sin Militancia		Esta opción indica el tipo de militancia a la que pertenece el militante.

➤ **CUResultadosPrimeraEntrevista:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	

Observaciones	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda la observación que ha recogido el psicólogo en la primera entrevista.
Continuación del proceso	Radiobutton	No	Si No		Este campo guarda si se le continuara el proceso al candidato o no.

➤ **CUResultadosAnálisisPsicologico:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Observaciones Publicas	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda la observación que ha recogido el psicólogo en el análisis psicométrico. Pero solo las observaciones que pueden ser del conocimiento de cualquier persona.
Observaciones Privadas	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda la observación que ha recogido el psicólogo en el análisis psicométrico. Pero solo las observaciones que pueden ser del conocimiento de los psicólogos que realizaron el proceso.
Continuación del proceso	Radiobutton	No	Si No		Este campo guarda si se le continuara el proceso al candidato o no.

➤ **CUResultadosVerificacion:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Continuación del	Radiobutton	No	Si No		Este campo guarda si se le continuara el proceso al

proceso					candidato o no.
---------	--	--	--	--	-----------------

➤ **CUPendientesAceptacion:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Aceptado	Radiobutton	Si	Aceptar Plaza		En este campo radiobutton se marca Aceptar plaza en el caso de que el candidato haya aceptado la plaza que se le propone.
Desertor	Radiobutton	Si			En este campo radiobutton se marca Desertor en el caso de que el candidato no haya aceptado la plaza que se le propone.

➤ **CUPlanificarDiasPsicometrico:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Fecha	Cadena de caracteres	No	Fecha en el formato dd/mm/aaa		En este campo se guardara una fecha en la cual se realizara pruebas psicométricas.
Cantidad	Numero entero positivos	No	Todos los dígitos del 0 al 9		En este campo se guardaran la cantidad de fichas que se procesara el día seleccionado.
Lugar	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el lugar donde se realizar la prueba de psicométrico

➤ **CUPlanificarPsicometrico:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Fecha	Cadena de caracteres	No	Fecha en el formato dd/mm/aaa		En este campo se guardaran las fechas ya seleccionadas en el caso de uso de Planificar días psicométrico.

➤ **CURegistrarAsistenciaalPsicometrico:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Fecha	Fecha	No	Fecha en formato dd/mm/aaaa		En este campo selecciona la fecha que se le desea buscar las fichas planificadas para la prueba psicométrica.

➤ **CUCrearAgenda:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Fecha	Cadena de caracteres	No	Fecha en el formato dd/mm/aaa		En este campo se guarda la fecha en la que se realizara la comisión correspondiente ya sea comisión de cuadro o comisión representativa.
Tipo de contrato	Cadena de caracteres	No	Contrato por Periodo de Prueba Contrato por tiempo indeterminado Contrato por tiempo determinado Contrato de Adiestramiento		En este campo se guarda el tipo de contrato el cual se le va a realizar a dicha persona (ya sea candidato o trabajador). En el caso de seleccionar Anexo es que al trabajador lo que se le va es a crear un Anexo al contrato.

			Formalización de la Designación Anexo		
--	--	--	---------------------------------------	--	--

➤ **CUAprobacionCandidatos:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Número de Acuerdo	Número entero	No	Todos los dígitos del 0 al 9 y cualquier combinación de estos		En este campo se guarda el número de acuerdo donde se aprueba la ficha seleccionada

➤ **CUCulminarFicha:**

Ultima Entrevista

**Tabla 4 Ultima Entrevista**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Número de Carnet de Identidad	Cadena de caracteres	No	Dígitos del 0 al 9, longitud de 11 caracteres		En este campo se registra el número de carnet de identidad del candidato.
Unidad	Cadena de caracteres de tipo select	No	Todas las unidades de la AGR		En este campo se selecciona la unidad de la aduana a la que se le va a buscar que plazas tiene vacante.
Sistemas Operativos	Cadena de caracteres	Si	Linux Microsoft Windows Mac OS Unix		Este campo almacena el tipo de sistema operativo del cual el candidato posee algún conocimiento.
Título	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		Este campo almacena el nombre del título en cualquier especialidad de la informática.

Centro	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el nombre del centro que fue el encargado de emitir dicho título
Lenguaje	Cadena de caracteres de tipo select	Si	Todos los lenguajes que existen.		En este campo se va guarda uno de los leguajes que domina el candidato.
Título	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el título que se le ha asignado en el lenguaje escogido anteriormente al candidato.
Centro	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el centro que fue el encargado de emitir el título del lenguaje seleccionado.
Leer	Booleano	Si	Leer		Este campo se marca en caso de que el candidato domine dicha habilidad en el lenguaje seleccionado.
Escribir	Booleano	Si	Escribir		Este campo se marca en caso de que el candidato domine dicha habilidad en el lenguaje seleccionado.
Hablar	Booleano	Si	Hablar		Este campo se marca en caso de que el candidato domine dicha habilidad en el lenguaje seleccionado.
Valoración	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda una valoración que realizan los psicólogos del candidato durante todo el proceso.

			Además de los dígitos del 0 al 9	
--	--	--	----------------------------------	--

**CUAsignarPlaza:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Dirigida a	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas		En este campo se registra a quien va dirigida dicha carta de aprobación.

**Módulo “Estructura y Composición”**
**➤ CUCrearUnidad:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Nombre	Cadena de caracteres	No	Todas las letras del alfabeto		Se registra el nombre de la unidad
Siglas	Cadena de caracteres	Si	Todas las letras del alfabeto		Se registra las siglas que van a representar dicha unidad. Ejemplo CADI, AAIJM.
Fecha de Resolución Vigente	dd/mm/aaaa	No			Se registra la fecha de la resolución del jefe de la Aduana General de la República que dio origen a la creación de la unidad.
Número de Resolución	Número	No	Los dígitos del 0 al 9 en cualquier		Se registra el número de la Resolución que

			combinación		dio origen a la creación de la Unidad.
Calle	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el nombre de la calle donde reside dicha unidad.
Entre	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha unidad.
Y	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha unidad.
Edificio	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número o nombre de dicho edificio.
No	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las		En este campo se guarda el número de la dirección de dicha unidad.

			vocales acentuadas. Además de los dígitos del 0 al 9		
Apto	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número del apartamento de dicha unidad.
Provincia	Cadena de caracteres	No	Pinar del Rio Ciudad de la Habana La Habana Matanzas Cienfuegos Villa Clara Sancti Spíritus Ciego de Ávila Camagüey Las Tunas Holguín Granma Santiago de Cuba Guantánamo Isla de la Juventud		En este campo se guarda el nombre de la provincia de dicha unidad.
Municipio	Cadena de caracteres	No	Todos los municipios		En este campo se guarda el nombre del

			correspondiente a la provincia seleccionada		municipio de dicha unidad.
Categoría	Cadena de caracteres	No	Ninguna 1RA Categoría 2DA Categoría 3RA Categoría		Se registra la categoría de la Unidad, en caso de que no tenga categoría el valor sería ninguna
Código de la unidad	Número	No	Dígitos del 0 al 9, longitud de 5 caracteres		En este campo se registra el identificador de la unidad.
Objetivos	Cadena de Caracteres	No	Todas las letras del alfabeto		Se registran los objetivos para los que se ha creado una nueva unidad.

➤ **CUCrearSubUnidad:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Nombre	Cadena de caracteres	No	Todas las letras del alfabeto		Se registra el nombre de la subunidad.
Siglas	Cadena de caracteres	Si	Todas las letras del alfabeto		Se registra las siglas que van a representar dicha subunidad.
Calle	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales		En este campo se guarda el nombre de la calle donde reside dicha subunidad.

			acentuadas.		
Entre	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha subunidad.
Y	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha subunidad.
Edificio	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número o nombre de dicho edificio.
No	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número de la dirección de dicha subunidad.

Apto	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número del apartamento de dicha subunidad.
Provincia	Cadena de caracteres	No	Pinar del Rio Ciudad de la Habana La Habana Matanzas Cienfuegos Villa Clara Sancti Spíritus Ciego de Ávila Camagüey Las Tunas Holguín Granma Santiago de Cuba Guantánamo Isla de la Juventud		En este campo se guarda el nombre de la provincia de dicha subunidad.
Municipio	Cadena de caracteres	No	Todos los municipios correspondientes a la provincia seleccionada		En este campo se guarda el nombre del municipio de dicha subunidad.
Objetivos	Cadena de	No	Todas las letras		Se registran los

	Caracteres		del alfabeto		objetivos para los que se ha creado una nueva subunidad.
--	------------	--	--------------	--	--

➤ *CUCrearUnidadOrganizativa:*

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no	
Nombre	Cadena de caracteres	No	Todas las letras del alfabeto		Se registra el nombre de la unidad organizativa
Siglas	Cadena de caracteres	Si	Todas las letras del alfabeto		Se registra las siglas que van a representar dicha unidad organizativa.
Calle	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el nombre de la calle donde reside dicha unidad organizativa.
Entre	Cadena de caracteres		Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha unidad organizativa.
Y	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicha unidad organizativa.
Edificio	Cadena de caracteres	Si	Todas las letras del vocabulario en		En este campo se guarda el número o

			español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		nombre de dicho edificio.
No	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número de la dirección de dicha unidad organizativa.
Apto	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número del apartamento de dicha unidad organizativa.
Provincia	Cadena de caracteres	No	Pinar del Rio Ciudad de la Habana La Habana Matanzas Cienfuegos Villa Clara Sancti Spiritus Ciego de Ávila Camagüey Las Tunas Holguín Granma		En este campo se guarda el nombre de la provincia de dicha unidad organizativa.

			Santiago de Cuba Guantánamo Isla de la Juventud		
Municipio	Cadena de caracteres	No	Todos los municipios correspondientes a la provincia seleccionada		En este campo se guarda el nombre del municipio de dicha unidad organizativa.
Objetivos	Cadena de Caracteres	No	Todas las letras del alfabeto		Se registran los objetivos para los que se ha creado una nueva unidad organizativa.

➤ **CUCrearGrupo:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Nombre	Cadena de caracteres	No	Todas las letras del alfabeto		Se registra el nombre del grupo de trabajo
Siglas	Cadena de caracteres	Si	Todas las letras del alfabeto		Se registra las siglas que van a representar dicho grupo de trabajo.
Calle	Cadena de caracteres	No	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda el nombre de la calle donde reside dicho grupo de trabajo.
Entre	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además		En este campo se guarda la entre que calle donde se

			de las vocales acentuadas.		encuentra dicho grupo de trabajo.
Y	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas.		En este campo se guarda la entre que calle donde se encuentra dicho grupo de trabajo.
Edificio	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número o nombre de dicho edificio.
No	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número de la dirección de dicho grupo de trabajo.
Apto	Cadena de caracteres	Si	Todas las letras del vocabulario en español, además de las vocales acentuadas. Además de los dígitos del 0 al 9		En este campo se guarda el número del apartamento de dicho grupo de trabajo.
Provincia	Cadena de caracteres	No	Pinar del Rio Ciudad de la Habana La Habana		En este campo se guarda el nombre de la provincia de dicho grupo de trabajo.

			Matanzas Cienfuegos Villa Clara Sancti Spiritus Ciego de Ávila Camagüey Las Tunas Holguín Granma Santiago de Cuba Guantánamo Isla de la Juventud		
Municipio	Cadena de caracteres	No	Todos los municipios correspondientes a la provincia seleccionada		En este campo se guarda el nombre del municipio de dicho grupo de trabajo.
Objetivos	Cadena de Caracteres	No	Todas las letras del alfabeto		Se registran los objetivos para los que se ha creado un nuevo grupo de trabajo.

➤ **CUCrearPlaza:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Nombre	Cadena de caracteres	No	Todas las letras del alfabeto		Se registra el nombre de la plaza
Estado	Cadena de caracteres	No	Ocupada Vacante Reservada		La plaza tiene un estado que por defecto al crearla será vacante.

Cargo	Cadena de caracteres		Todos los cargos validos para la AGR		Este campo registra el nombre del cargo correspondiente a la plaza
Persona					que se crea. Estos cargos son los establecidos por la AGR.

➤ **CUAsignarPersonalidadJurídica:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	
Nombre	Cadena de caracteres	No	Todas las letras del alfabeto		Se registra el nombre de la unidad
Número de la resolución del Ministerio de Economía y Planificación	Número entero	No	Todos los dígitos desde el 0 al 9		Se registra el numero de la resolución que es la encargada de asignar la personalidad jurídica a la unidad.
Fecha de vigencia de la resolución		No	dd/mm/aaaa		

➤ **CURealizarConversionesdePlazas:**

Nombre	Tipo	Puede ser nulo	Restricciones		Descripción
			Clases Válidas	Clases no Válidas	

Nombre	Cadena de caracteres	No	Todas las letras del alfabeto		Se registra el nombre de la plaza
Estado	Cadena de caracteres	No	Ocupada Vacante Reservada		La plaza tiene un estado que por defecto al crearla será vacante.
Cargo	Cadena de caracteres		Todos los cargos validos para la AGR		Este campo registra el nombre del cargo correspondiente a la plaza
Persona					que se crea. Estos cargos son los establecidos por la AGR.