



Centro de Investigación en Matemáticas, A.C.

# Maestría en Ingeniería de Software



## **EL USO DEL ORGANIGRAMA COMO BASE PARA LA ARQUITECTURA DE ENTIDADES GUBERNAMENTALES**

**Hugo Aarón Rodríguez Villarreal**

**Cuauhtémoc Lemus Olalde**

## INTRODUCCIÓN

Lo primero que comprendí cuando empecé a estudiar la Maestría en Ingeniería de Software fueron dos cosas: primero que no existe la “bala de plata” que pueda ayudarte a resolver todos los problemas en el desarrollo de software, y segundo que (esta reflexión fue la que motivo el desarrollo de este tema) a diferencia de otras ingenierías como mecánica, electrónica, civil o la propia arquitectura (me refiero a la profesión) donde ya se tienen bien definidos los procesos, técnicas y métodos; los cuales pueden expresar de manera “clara” los requerimientos de un proyecto.

En la construcción de una casa donde se involucran arquitectos, albañiles, electricistas, plomeros, diseñadores, ingenieros civiles, la comunicación entre ellos es en base a planos que expresan los requerimientos del cliente e indican sin ambigüedades lo que cada uno de los involucrados debe hacer.

Esta falta de “planos” en la ingeniería de software es la que muchas veces ocasiona que los proyectos fallen, ya que no se expresa ni se entiende de manera adecuada las necesidades del cliente [8]. Ya que la mayoría de las veces la arquitectura es irrelevante para los clientes, solamente sirve para el desarrollo [4].

Por ello la idea de poder utilizar un “plano” que pueda ser entendido por todos los involucrados en el proceso desde el cliente, pasando por el ingeniero de requerimientos, el arquitecto, el diseñador, el programador, el ingeniero de pruebas, el implementador y los usuarios. Y este plano puede ser (como base) el organigrama de una organización.

## ARQUITECTURA

Los elementos comunes en las definiciones de arquitectura de software son componentes y las relaciones entre ellos. Como lo podemos ver en la definición de Bass, Clements y Kazman [12],

*“La arquitectura de software de un programa o sistema informático es la estructura o estructuras del sistema, que incluyen elementos de software, las propiedades visibles externamente de esos **elementos**, y las **relaciones** entre ellos”;*

así como, en la definición de la IEEE[13],

*“La arquitectura se define por la práctica recomendada fundamentales como la organización de un sistema, plasmado en sus **componentes**, sus **relaciones** entre sí y el medio ambiente y los principios que rigen su diseño y evolución”.*

La arquitectura de software no es nada más componentes y relaciones, Garlan and Shaw, 1993: en un “seminal paper on software architecture” sugieren que existen “issues” en el diseño de arquitecturas.



*“...más allá de los algoritmos y estructuras de datos de la computación, el diseño y especificación de la estructura general del sistema se perfila como un nuevo tipo de problema. Cuestiones estructurales en cifras brutas incluyen la organización y estructura de control mundial, los protocolos de comunicación, sincronización y acceso a los datos, asignación de funciones a los elementos de diseño, distribución física, la composición de elementos de diseño, la ampliación y el rendimiento, y la selección entre alternativas de diseño”.*

Una arquitectura debe expresar más que simplemente componente e interacciones. La construcción de cualquier estructura está bien definida por su arquitectura [5], una arquitectura bien definida permite entender la estructura que se construye, planear su construcción y estimar los costos.

En el campo de “business architecture” el modelado de una arquitectura de negocios debe responder varias preguntas [5]:

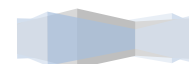
- ¿Cómo interactúan los diferentes actores?,
- ¿Qué actividades son parte de su trabajo?,
- ¿Cuál es la objetivo de su trabajo?,
- ¿Qué otras personas, recursos o sistemas están involucrados los cuales no se muestran como actores de este sistema?,
- ¿Qué reglas gobiernan las actividades y las estructuras?,
- ¿Existe otra manera de que los actores sean más eficientes?.

A demás de todo lo anterior planteado, una buena arquitectura en el modelado de negocios debe [5]:

- Captura el negocio de manera correcta y verdadera
- Se enfoca en el proceso principal
- Representa una vista concienzuda de la operación del negocio
- Se adapta fácilmente a los cambios y extensiones
- Es fácil de entender y permite la comunicación entre los diferentes involucrados

## ¿POR QUÉ EL ORGANIGRAMA?

Historia de dos fracasos en la adquisición de programas de software. Respetando a la privacidad y el derecho legítimo a la confidencial de la información me voy a referir a las organizaciones donde se presentaron dichas experiencias como la empresa P y C respectivamente. La empresa P es un gran corporativo la cual adquirió un software de unos \$700,000.00 dólares, pero este software no contaba con los reportes que los directivos necesitaban, así que el área de sistemas de la empresa tuvo que programar dichos reportes para satisfacer esa necesidad. Por otro lado, la empresa C, cuenta con múltiples sucursales alrededor del país, y dicha organización adquirió un software de más de \$1,000,000.00 de dólares, el cual le permite el control de las operaciones diarias de la empresa, pero la información requerida por los directivos para tomar decisiones de tipo gerencial no es



proporcionada por dicho software, así que, el personal de mandos medios tiene que utilizar hojas de cálculo para procesar la información, y así proporcionar la información necesaria por los altos directivos.

La falta de visión global de los ingenieros de software puede generar en sistemas que no sean del todo aprovechables por la empresa. Como se mencionó anteriormente, las empresas P y C cuentan con sistemas que de alguna manera les sirven para la operación de cierta parte de la empresa, pero los altos directivos de la organización no tienen acceso a la información de dichos programas para la toma de decisiones, ver figura 2. Esto se debe a que la mayoría de las veces, los desarrolladores de software se enfocan en los procesos y en las actividades propias de la organización y no se enfocan en las necesidades de los directivos y de las demás áreas de la organización. De aquí surge la necesidad de un elemento que sirva para visualizar a toda la organización y que a la hora del desarrollo se esté seguro que se están contemplando todas las partes importantes de la organización y todas las interacciones posibles.

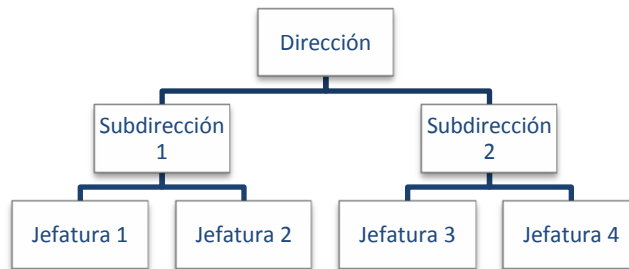


Figura 1. Ejemplo de un organigrama

Una forma de ver a una organización como un todo es con su organigrama [1]. Un organigrama es la representación gráfica de la estructura organizativa de una empresa u organización; es un modelo abstracto y sistemático, que permite obtener una idea uniforme acerca de la estructura formal de una organización, es la representación formal de toda la organización. [7] En el desarrollo de sistemas empresariales, se deben contemplar todos los aspectos de la empresa e involucrar a toda la organización, y un organigrama puede ofrecernos un buen mapa para empezar.

Los organigramas pueden clasificarse entre cuatro grandes grupos, por su naturaleza, por su ámbito, por su contenido y por su presentación. [14] En este último existen cuatro tipos de organigramas, entre ellos están los lineales o verticales ver la figura1, que representan en forma vertical la línea de autoridad de la organización de arriba hacia abajo; los horizontales representan las unidades de izquierda a derecha, los mixtos utilizan una



combinación de vertical y horizontal para ampliar la graficación y de bloque son una variante de los verticales y su particularidad es que integran más unidades en espacios más reducidos..

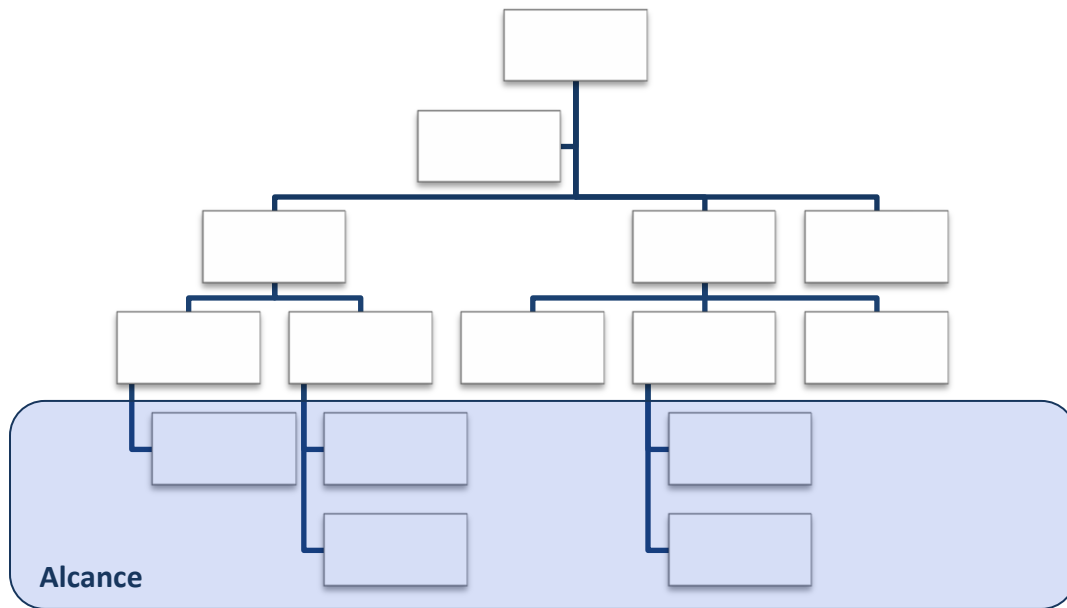


Figura 2. Alcance de los sistemas

El organigrama ¿sirve como base para la arquitectura de software?. Existen varias definiciones de arquitectura de software, pero la mayoría coincide en elementos clave de una arquitectura; es una vista abstracta de la estructura del sistema, de los componentes y de sus relaciones [2]; entre otras muchas características que se pueden citar. Si vemos a un organigrama como una vista abstracta de una organización, representa a los componentes (departamentos) que la forman y las relaciones que tienen entre ellos; lo podríamos tomar como un mapa o plano arquitectónico que representa todos los componentes que debe contener el sistema.

Si bien es cierto, el organigrama como tal no es una arquitectura. Ya que dicho organigrama no dirá ¿Cuál es la naturaleza de los componentes?, ¿Cuál es el significado de su separación?, ¿Cuál es el significado de las relaciones?, ¿Cuál es el significado de su forma? [2]. Si se podría usar como base o como un elemento de una arquitectura. Permitiendo ser una guía a gran escala del desarrollo de software.

Según Clements [2], existen tres razones fundamentales de por qué es importante una arquitectura de software. Permite la comunicación entre los involucrados (un organigrama permite la comunicación entre el cliente y el equipo de desarrollo), permite la toma temprana de decisiones de diseño (al utilizar un organigrama como base para el diseño, se pueden inferir rápidamente muchas necesidades y limitantes que contendrá el sistema) y es una abstracción del sistema transferible (como tal el organigrama es una abstracción de la organización que puede mostrar una abstracción del sistema).



El organigrama permite la comunicación entre los involucrados. El primer involucrado es el cliente y usar un organigrama como la base de la arquitectura de software, es hablar en el mismo lenguaje del cliente y no utilizar diagramas complejos que no son fácilmente entendibles.

El organigrama permite tomar decisiones tempranas de diseño. Ya que nos muestra a todos los involucrados en la organización, a todas las áreas de operación de la organización y con un vistazo se puede crear un diseño de todo lo que contendrá el sistema.

El organigrama es un diseño abstracto transferible. Puede ser utilizado en cualquier fase del desarrollo para tomar decisiones, como la comunicación entre componentes, o en el diseño físico que nos puede indicar que tanta capacidad requiere el servidor u otros elementos necesarios.

Se pueden hacer buenas arquitecturas usando como base a un organigrama. Garlan [3] menciona que una arquitectura debe abarcar toda la organización y contener todos los componentes y su interacción. Todos los autores coinciden con que la arquitectura es la representación de componentes y sus relaciones, visto de un nivel abstracto alto, que debe representar todo el sistema [6]. Un organigrama puede satisfacer esas características ya que abarca toda la organización y la comunicación de todos sus elementos.

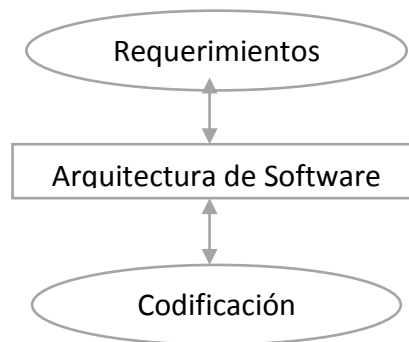


Figura 3. Arquitectura de software como puente. Garlan [3]

Un organigrama puede ser un vehículo para comunicar de manera más efectiva los requerimientos y contribuir al desarrollo del sistema. Garlan [3] menciona que una arquitectura es un puente entre los requerimientos y la codificación ver figura 3. Y el organigrama puede ser un puente entre el cliente y los requerimientos, de tal manera que se puede realizar una arquitectura enfocada a las necesidades del cliente y por consiguiente que el desarrollo del software este alienado a las necesidades del cliente ver figura 4.



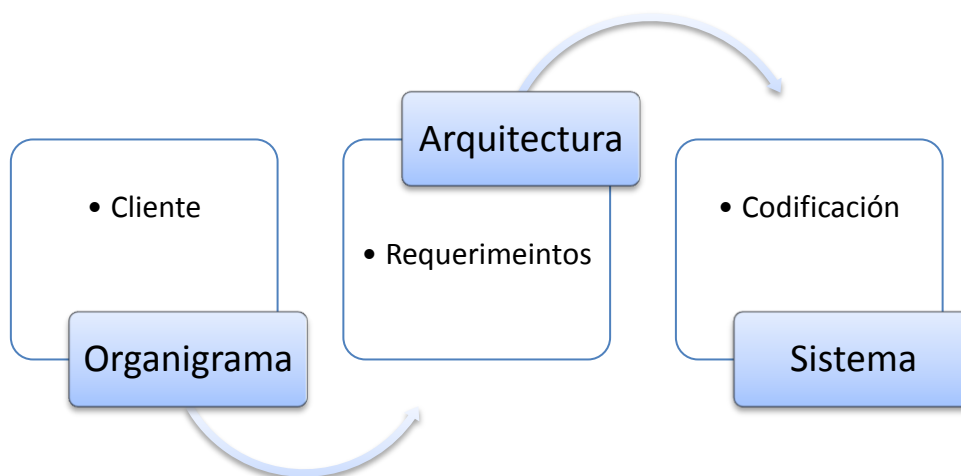


Figura 4. El organigrama como puente.

El organigrama cuenta con elementos suficientes para usarse como base en una arquitectura. Permite identificar los elementos y su comunicación y esto es parte fundamental en una arquitectura de software. Permite usarlo como un mapa para la guía en el desarrollo y eso es lo que busca una arquitectura. Permite el entendimiento entre los involucrados y precisamente una arquitectura pretende ser un lenguaje común entre todos los involucrados en el desarrollo de software.

Debemos estar conscientes de que un negocio (organización) es un sistema complejo [6]. Es una organización jerárquica de departamentos y sus funciones, esas funciones muchas veces no son exclusivas de un departamento sino que cruzan de manera horizontal entre departamentos. Para ello se utiliza el organigrama o diagrama organizacional como documentación, aunque este documento no representa los procesos de negocio [11].

La función principal de una arquitectura es la de comunicación entre los diferentes actores. Entender una arquitectura es factor primordial para el existir del desarrollo, esta debe ser comprendida por todas las personas involucradas y no debe representar un esfuerzo para ninguna de ellas el entenderla, ya que esto es lo que ocasiona re-trabajo en los desarrollos. Por ello basar la arquitectura en un organigrama puede ser un medio de comunicación eficaz entre los diferentes involucrados en el desarrollo y principal mente con el cliente.

## ¿POR QUÉ GOBIERNO?

Las entidades gubernamentales tienen la estructura orgánica más arraigada en sus raíces. Las entidades gubernamentales tienen en la ley que las crea, ya plasmada su estructura orgánica y por consiguiente las funciones de estas entidades están plasmadas en la ley o



en los reglamentos. Lo que genera que las entidades gubernamentales se apeguen a su organigrama y por consiguiente refleje de manera puntual la operación de cada entidad.

La cultura organizacional en los gobiernos es fundamental para la operación del organigrama. En el ámbito gubernamental, las líneas jerárquicas y de comunicación que se expresan en los organigramas son más respetados que en otras instituciones no gubernamentales. Por lo que la operación de las entidades de gobierno está basada en su estructura organizacional.

La mayoría de las organizaciones no tienen documentados sus procesos [5]. Si consideramos que las empresas privadas en su mayoría no cuentan con sus procesos bien documentados; las organizaciones gubernamentales, en su mayoría no cuentan con procesos definidos, y a eso le agregamos, que las empresas de gobierno se rigen más por la situación política, por el presupuesto, por el momento electoral, por los deseos del titular de la dependencia, por las políticas propias del gobernador o presidente en turno; la mayoría de los procesos no se respetan ya existen otras “prioridades”.

Para desarrollar un sistema en gobierno hay que considerar los factores jerárquicos. La mayoría de las actividades cuentan con el visto bueno del superior jerárquico y en su caso tiene la última palabra en lo referente al curso del proceso, si no existe ninguna contrariedad los procesos funcionan adecuadamente, pero en caso de “necesidades del servicio” pueden cambiar las prioridades.

## PROPUESTA

### El cubograma

Las diferentes posibilidades que tiene el organigrama como la base para el diseño de la arquitectura de un sistema son varias pero nada más me voy a enfocar en las tres primeras etapas del proceso de desarrollo. Antes de mencionar las etapas es necesario mencionar la necesidad de utilizar algo que yo llamo “**cubograma**” como la base de esta propuesta. La primera utilidad que tiene el organigrama en el proceso de desarrollo es para la obtención de requerimientos, ya que sirve como una referencia para guiarnos en el proceso de requerimientos. La segunda utilidad una vez que obtuvimos los requerimientos es para complementar la estimación. Y por último para el diseño de la arquitectura.

¿Qué es el *cubograma*?. En esta propuesta el *cubograma* es el elemento esencial para conceptualizar el componente principal para el diseño basado en el organigrama. El cubograma se basa en las funciones básicas que representa un organigrama; así como, en los medios de comunicación que existen dentro de la organización y que están representados por el organigrama.

El *cubograma* es una representación de la comunicación dentro de la organización, complementada con la comunicación hacia el exterior de la misma. Representa la comunicación directa entre el superior y sus subordinados, entre miembros del mismo nivel





jerárquico, entre personal de apoyo (staff) y personal operativo, la operación interna del propio departamento y la comunicación hacia el exterior.

### El cubo

El *cubo* es la representación de los departamentos o áreas del organigrama dentro del cubograma, cada cubo cuenta con 6 lados o caras, cada lado representa un canal de comunicación. Las dos últimas caras no se encuentran representadas en el organigrama pero son el complemento para completar el diseño de la arquitectura.

1. Cara superior, representa la comunicación hacia arriba, hacia el superior jerárquico.
2. Cara derecha, representa la comunicación hacia elementos del mismo nivel jerárquico.
3. Cara izquierda, representa la comunicación hacia elementos staff.
4. Cara inferior, representa la comunicación hacia abajo, hacia los subordinados.
5. Cara posterior, representa la comunicación interna.
6. Cara frontal, representa la comunicación hacia el exterior.

Cabe mencionar que aunque en el organigrama se representa la comunicación formal, en el *cubograma* además se representa el intercambio de información.

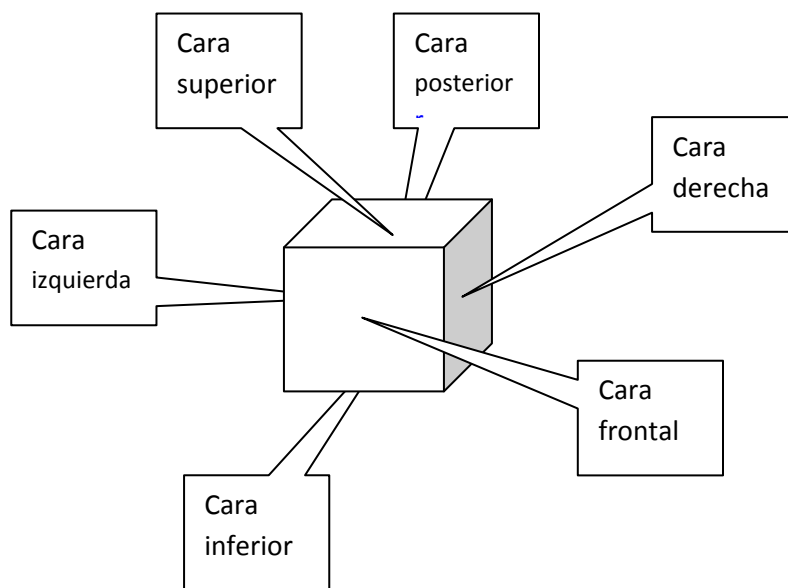


Figura 5. Representación de un “cubo”.

Cabe aclarar que el cubo, puede representar una aplicación, una base de datos, una interfaz (usuario), un subsistema o un componente.

Se ejemplifica la representación del cubograma dentro del organigrama, la figura 6 es un organigrama común, con una dirección, un departamento staff, dos sub direcciones y cada una de ellas con dos jefaturas. En la figura 7 se representa un cubograma de la subdirección 1.



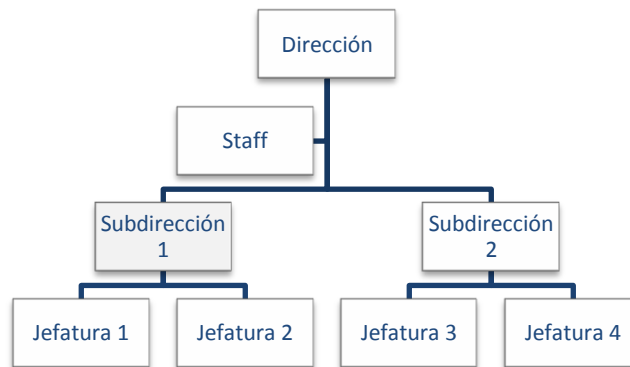


Figura 6. Organigrama con departamento staff.

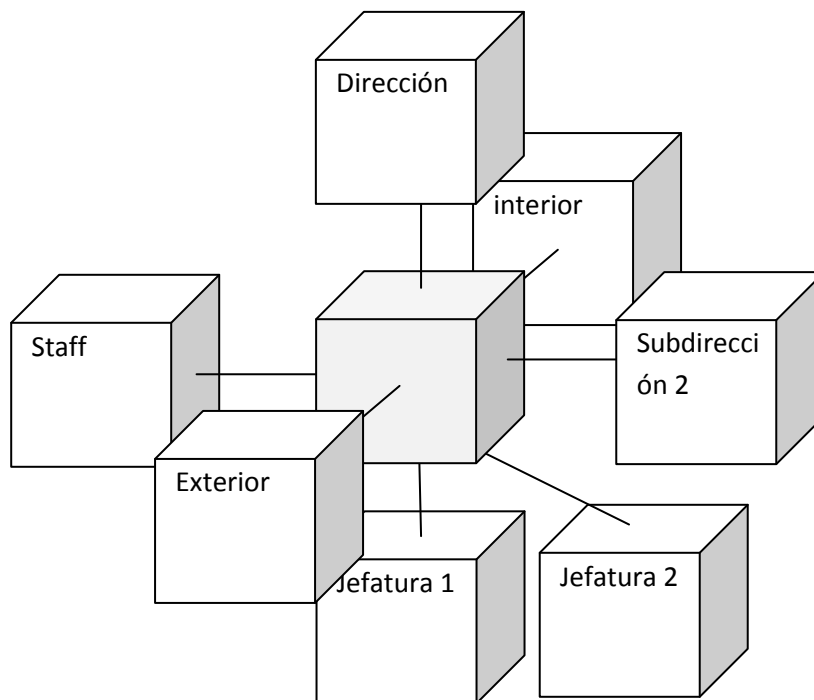


Figura 6. Cubograma de la subdirección 1.

El cubograma es la base para el diseño de la arquitectura, de donde se obtienen los componentes que se van a desarrollar; así como, de todas las comunicaciones y conexiones del mismo.

El cubograma se inicia con el “cubo” base de la aplicación, conectada a las áreas dentro de la organización, una vez identificado los “cubos” conectados, se debe realizar el análisis de cada “cubo” a las demás áreas de la organización, pero dentro del ámbito del mismo sistema.



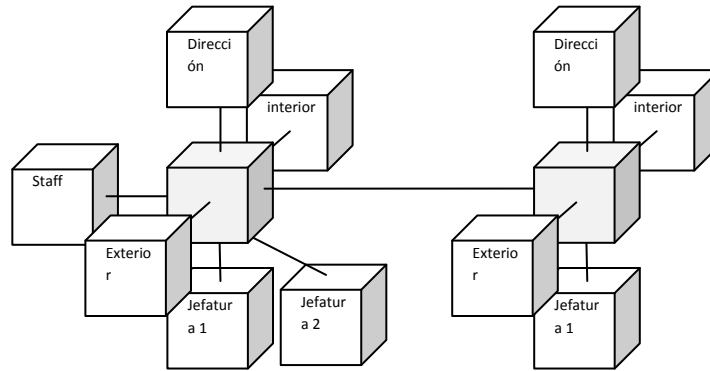


Figura 7. Cubograma de un sistema completo

**Elementos a considerar en el diseño del cubograma.**

Existen 7 elementos que se consideran para el uso del cubograma, estos elementos se aplican a cada una de las caras del cubo, estos elementos son medios de comunicación o interfaces entre los cubos del cubograma, y son los básicos que se deben considerar además de los que surjan durante el proceso de requerimientos y de diseño.

Existen dos tipos de interfaces, las básicas y las chart (de organigrama). Estas interfaces están basadas en los aspectos básicos de comunicación y colaboración dentro de la estructura orgánica de la organización.

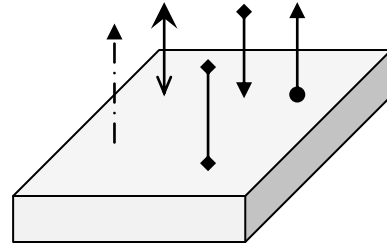
- Interfaces básicas
  - Identificación ●————→
  - seguridad ◆————◆
- Interfaces chart
  - Instrucción ◆————→
  - Autorización ←————→
  - Conexión ●————●
  - Reporte - - - - -→
  - Colaboración ←————→

Cada una de las caras del cubo implementa uno o varias interfaces, dependiendo del tipo de cara que se trate. Todas utilizan la interfaz de identificación, cuando es requerido utilizan la de seguridad, aunque pueden utilizar cualquiera de las interfaces, es recomendable como mínimo las que se muestran a continuación,

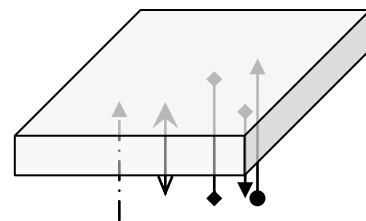


aunque estas no son privativas pueden existir un sinfín de interfaces adicionales dependiendo del ámbito del sistema.

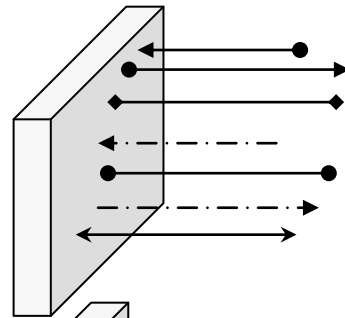
- La cara superior
  - Provee Identificación
  - Provee Seguridad
  - Requiere instrucción
  - Requiere autorización
  - Provee reporte



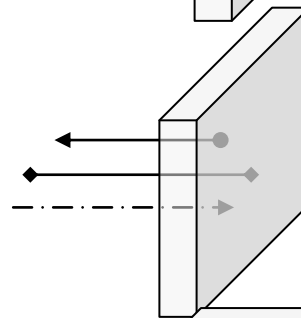
- La cara inferior
  - Requiere identificación
  - Requiere seguridad
  - Provee instrucción
  - Provee autorización
  - Requiere reporte



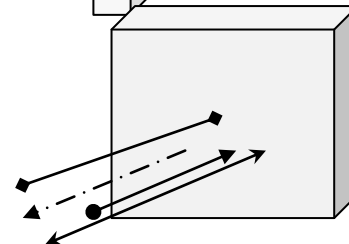
- La cara derecha
  - Provee y requiere identificación
  - Provee y requiere seguridad
  - Provee y requiere conexión
  - Provee y requiere colaboración
  - Provee y requiere reporte



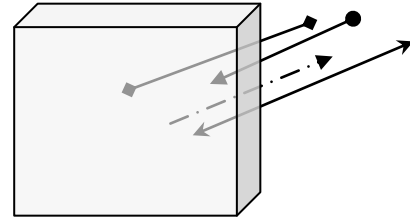
- La cara izquierda
  - Requiere reporte
  - Requiere identificación
  - Requiere seguridad



- La cara frontal
  - Requiere identificación
  - Requiere seguridad
  - Provee reporte
  - Requiere colaboración

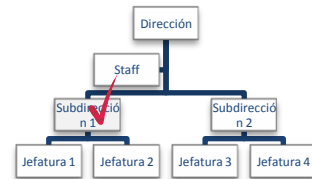


- La cara posterior
  - Requiere identificación
  - Requiere seguridad
  - Provee reporte
  - Requiere colaboración

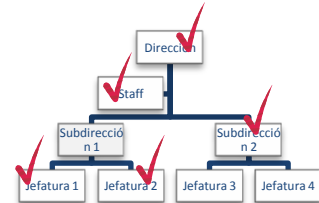


**Pasos para el diseño del cubograma**

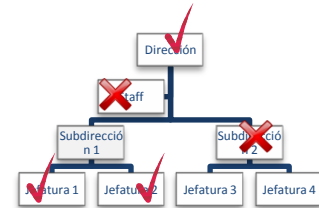
1. Identificar dentro del organigrama el área o departamento donde se va a desarrollar el software.



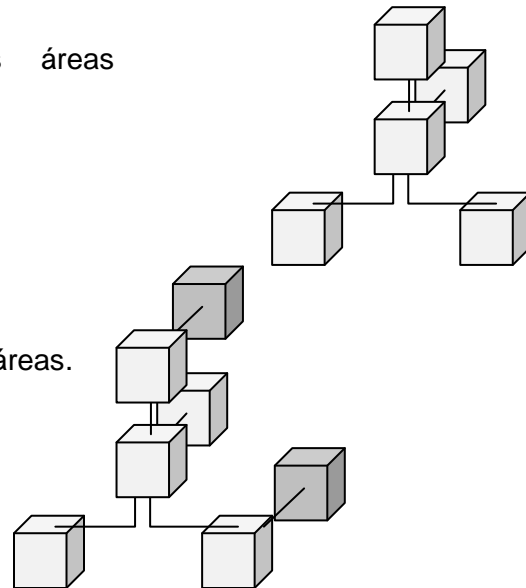
2. Identificar los departamentos con los que tiene relación dicho departamento.



3. Descartar las áreas que no tengan injerencia en el proceso o ámbito del sistema.



4. Construir el cubograma con las áreas resultantes.



5. Extender el cubograma a las diferentes áreas.



6. Identificar por cada cara cuales interfaces se deben implementar.

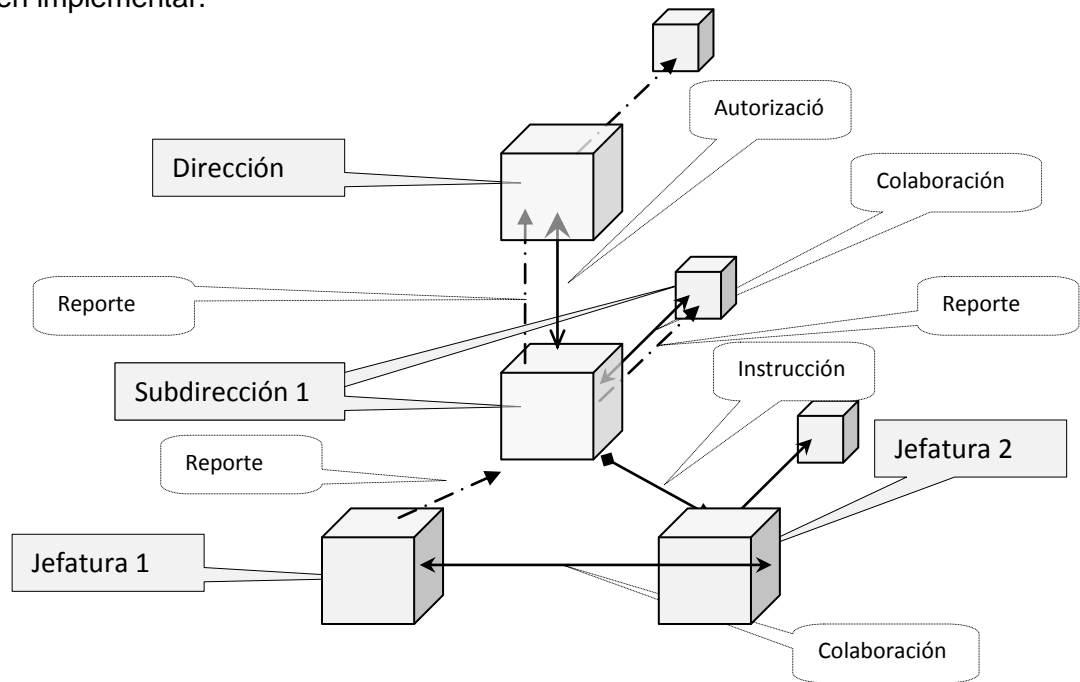


Figura 7. Cubograma de subdirección 1 con interfaces.

La figura 7 muestra un cubograma de un sistema, el cual requiere la autorización del nivel superior para poder iniciar; una vez autorizado se da una instrucción al nivel inferior (jefatura 2) para que conjuntamente con la jefatura 1 realice una actividad; una vez terminada, la jefatura 1 realiza un reporte, el cual es trabajado por la subdirección 1 y finalmente entregado a la dirección.

### Consideraciones en el diseño del “cubograma”

El cubo principal, representa la computación de la aplicación, los cubos adicionales pueden representar una base de datos, otra aplicación o simplemente una pantalla de captura o de reporte.

Cada cubo principal debe poseer su propia base de datos, ya que se considera que los cubos son independientes de los demás elementos del sistema. Esto permite mayor flexibilidad en las modificaciones, actualizaciones y agregados en el sistema.

Las interfaces propuestas, no limitan el uso de otras interfaces, e incluso pueden omitirse, lo que se propone es para que exista unificación entre la comunicación de los cubos y el cubo principal, para posibles modificaciones al sistema.



## Diseño de la arquitectura usando como base el cubograma

Usar vistas es la mejor forma de comunicar las necesidades a los diferentes involucrados. [11] una vista es una representación de un conjunto de elementos de sistema y las relaciones asociadas a ellos. Como el cubograma es una representación de elementos y sus relaciones, la mejor aproximación es el uso de las vistas para representar la arquitectura.

Para el diseño de las vistas se usará UML, ya que es uno de los estándares más usados por la industria.

Primero se transforma la representación del cubograma en un componente usando UML components. Ya que un componente nos sirve para representar los elementos principales de ejecución, elementos de almacenaje como clientes, servidores, filtros y bases de datos. Los conectores representan la interacción entre los componentes como cliente servidor, pipes and filters. Las interfaces la interacción entre los componentes y su entorno. Usaremos los elementos de UML components para transformar el cubograma en un diagrama de UML components.

Usaremos dos elementos para la representación de los componentes, [11], puertos e interfaces, los puertos son como las interfaces, pero cada puerto es diferente según su clasificador, el clasificador indica la multiplicidad de un puerto; las interfaces representan la conexión del componente con otros componentes.

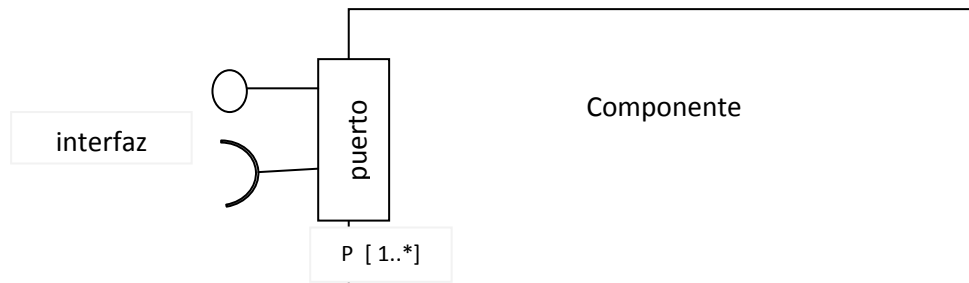


Figura 8. Componente con puertos e interfaces.



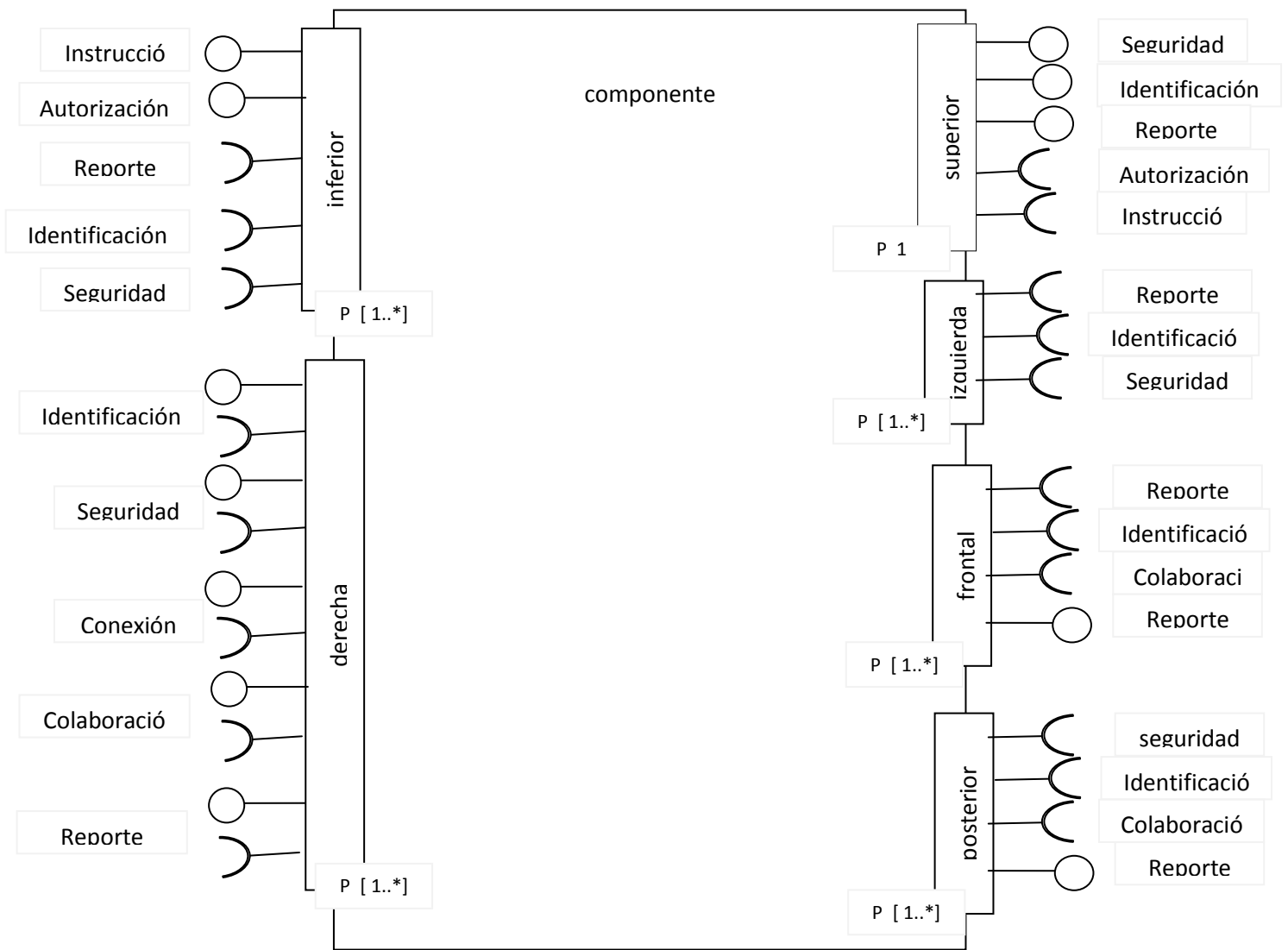


Figura 9. Componente con puertos e interfaces básicos del cubograma.





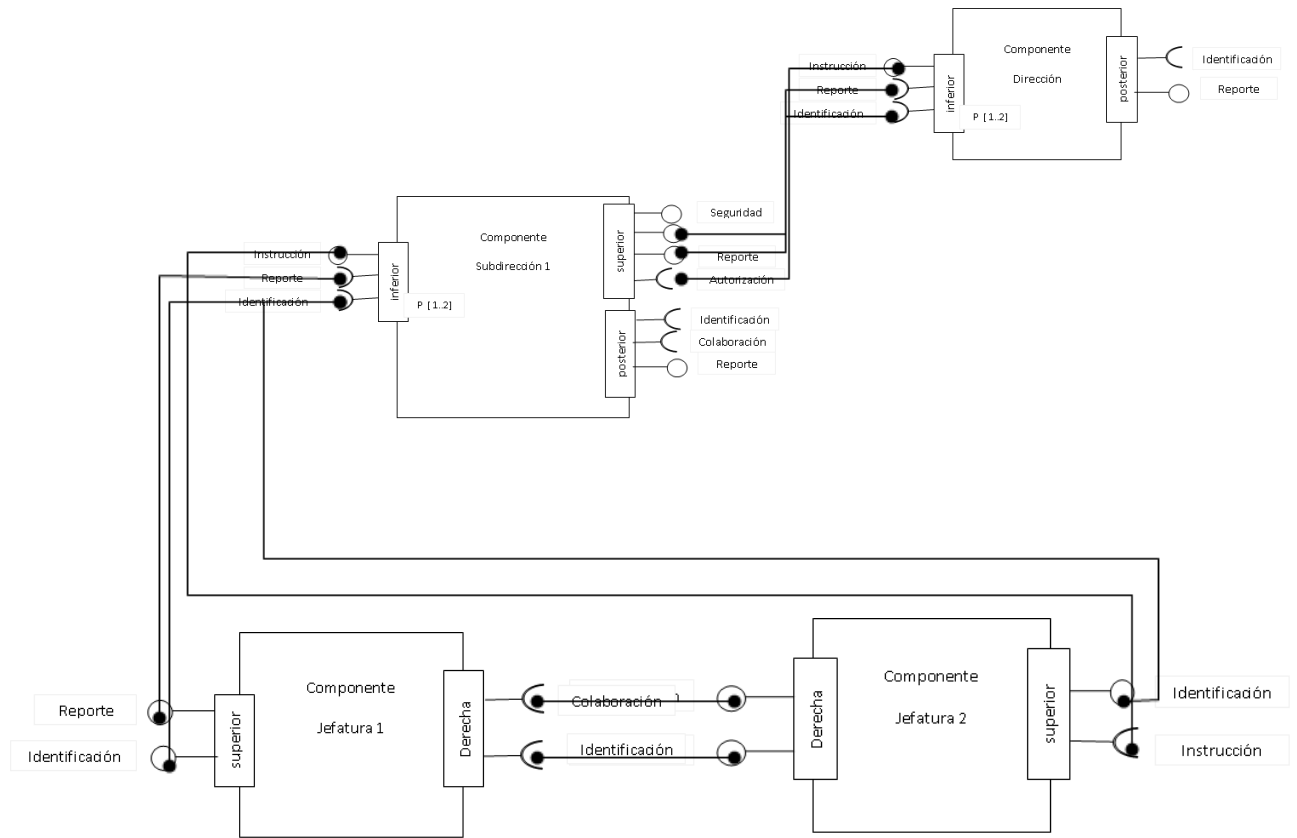


Figura 10 Diagrama de componentes de la figura 7.

## CASO PARACTICO

El caso práctico se está desarrollando en una dependencia de gobierno del estado de Zacatecas, se hará un desglose de las fases de requerimientos y diseño para ejemplificar el uso del cubograma.

### Sistema de Catálogo de Servidores

#### Requerimientos iniciales

“Se desea una programa para administrar el catalogo de servidores, ya que actualmente cada dirección cuenta con su propio catálogo o utilizan una versión que ya no está validada.” El catalogo de servidores es la lista de servidores públicos autorizados por la ley avalados por la Oficialía Mayor de Gobierno para pertenecer al Servicio profesional de Carrera del Estado.

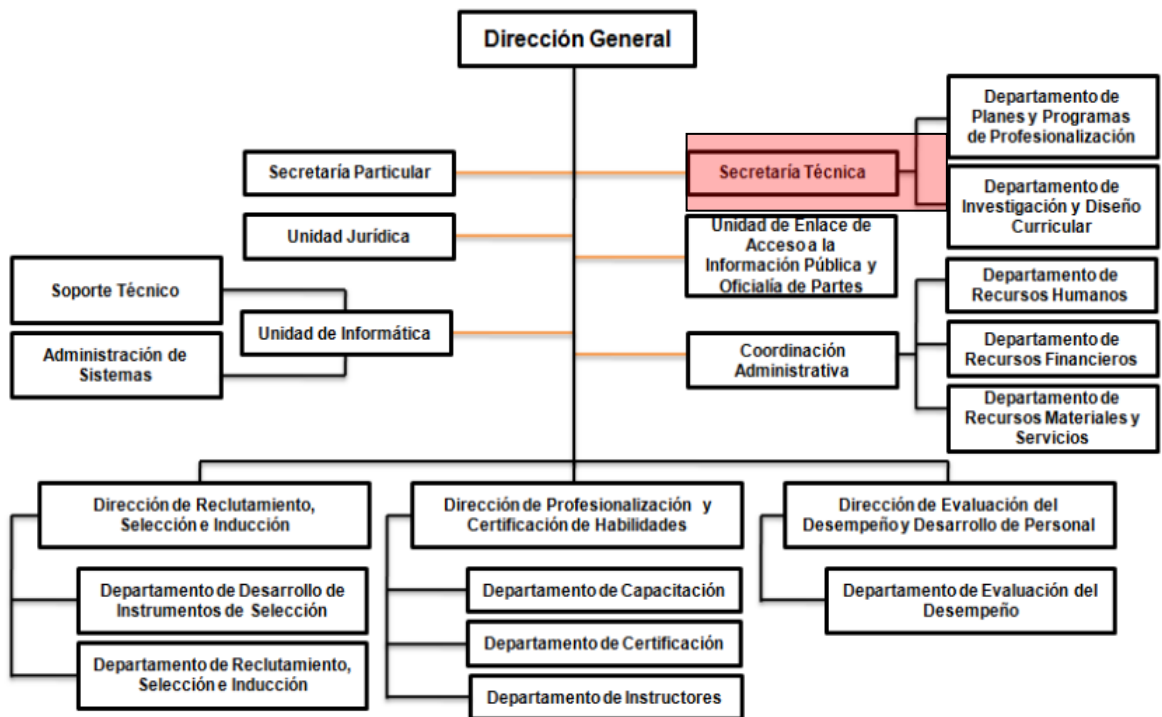
Esta lista se modifica mes con mes, y es proporcionada por la oficialía mayor por medio de un archivo de Excel.



Además se requiere que las direcciones actualicen la información de cada servidor, respecto a los procesos que cada una de ellas realiza; es importante que toda la información la pueda consultar el Director General en cualquier momento.

**Paso 1. Identificar dentro del organigrama el área o departamento donde se va a desarrollar el software.**

El programa será operado por la Secretaría Técnica de la dependencia.

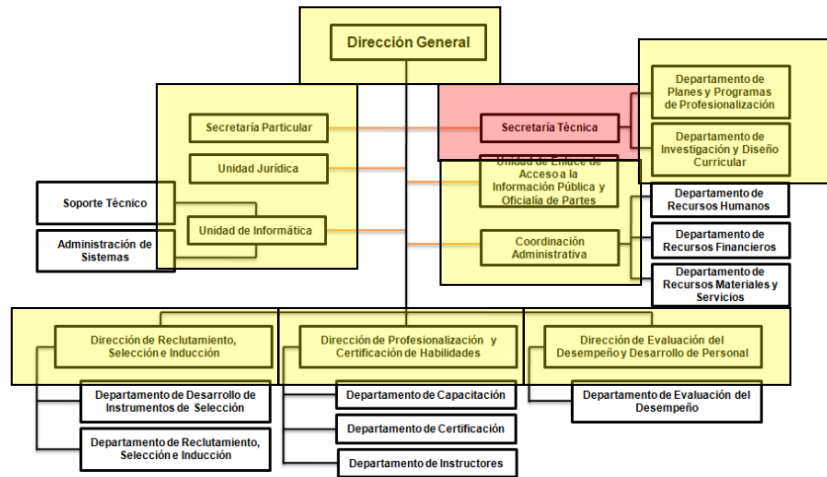


**Organigrama de la Dependencia**



**Paso 2. Identificar los departamentos con los que tiene relación dicho departamento.**

Se consulta directamente el organigrama para ver los departamentos que están relacionados con la Secretaría Técnica.



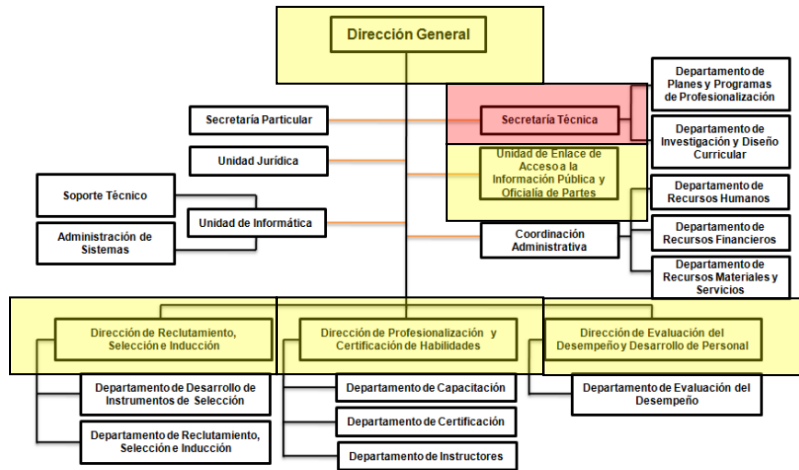
**Departamentos relacionados directamente**

**Paso 3. Descartar las áreas que no tengan injerencia en el proceso o ámbito del sistema.**

De las áreas que se encuentran relacionadas, ¿cuál de ellas tiene injerencia en el sistema? Y ¿cuál es su interés en el mismo?

- La oficialía de partes y unidad de acceso a la información, necesita información para el portal de transparencia.
- La dirección general requiere tener acceso a la información de todas las direcciones.
- La dirección de reclutamiento, necesita la lista de los servidores públicos y actualizar información profesional de los servidores.
- La dirección de profesionalización, necesita la lista de los servidores públicos, y actualizar quienes ha llevado alguna capacitación.
- La dirección de evaluación, necesita la lista de servidores públicos y registrar los resultados de las evaluaciones del desempeño semestral.

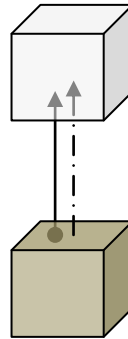




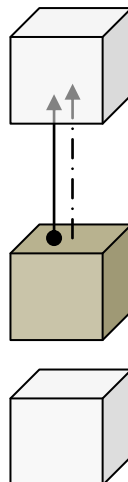
**Departamentos relacionados con la aplicación**

**Paso 4. Construir el cubograma con las áreas resultantes.**

- a. Cara superior
  - i. El director consulta la información generada por las direcciones

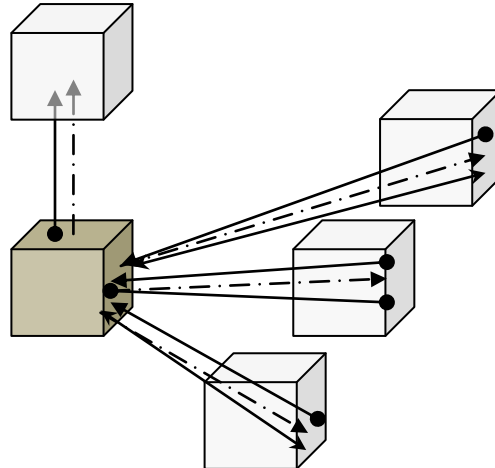


- b. Cara inferior
  - i. Los subordinados no tienen acceso a la aplicación, por lo tanto no aplica.



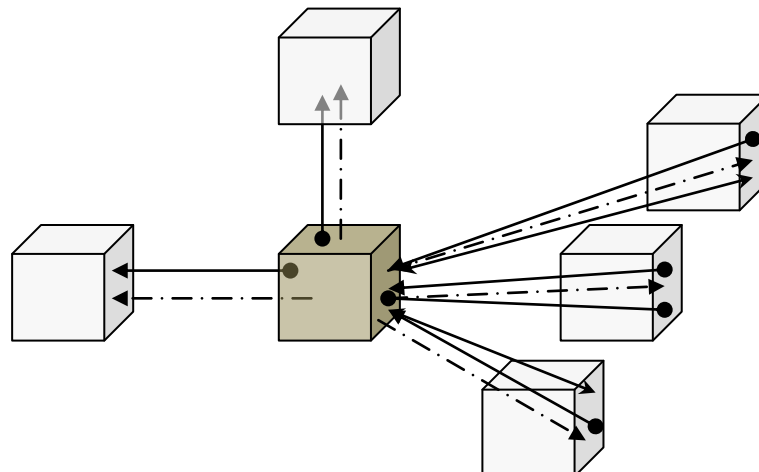
c. Cara derecha

- i. En este caso como el departamento es una unidad staff, los miembros del mismo nivel serian las direcciones.
- ii. Las tres direcciones actualizan a la aplicación
- iii. La dirección de evaluación, ya cuenta con un sistema informático que contiene la información, por lo que el sistema se conecta a la base de datos

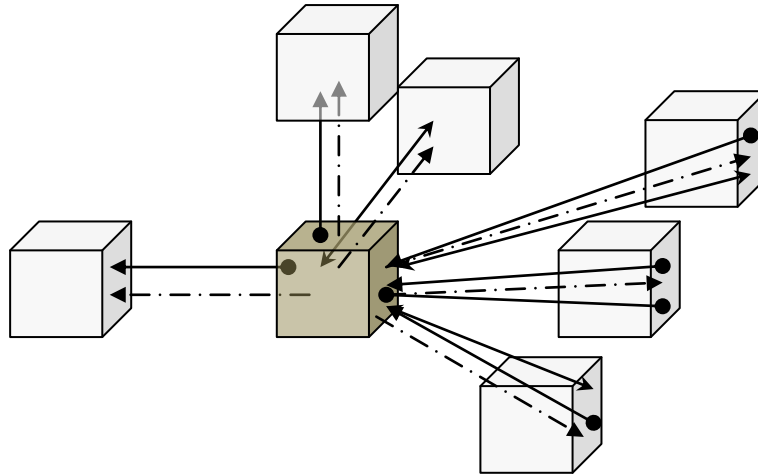


d. Cara izquierda

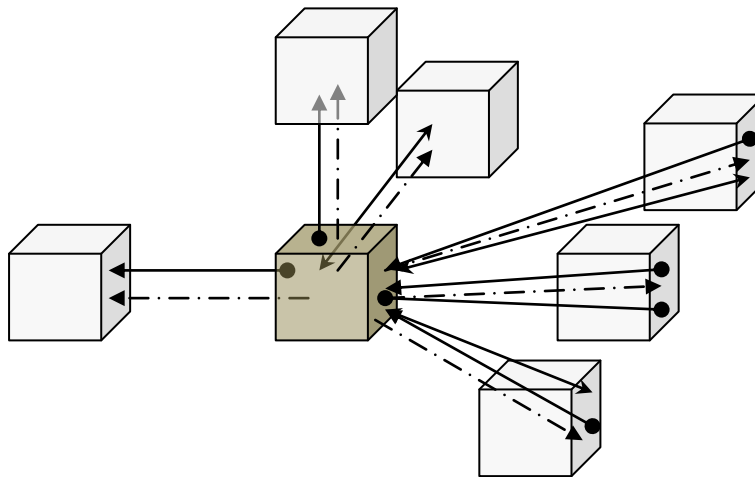
- i. Con unidades de staff se relaciona con la oficialía de partes enviando un reporte para el portal de transparencia



- e. Cara posterior
  - i. La secretaría técnica actualiza la aplicación con información proveniente de oficialía mayor.



- f. Cara frontal
  - i. Como la aplicación no tiene acceso al exterior, no aplica.

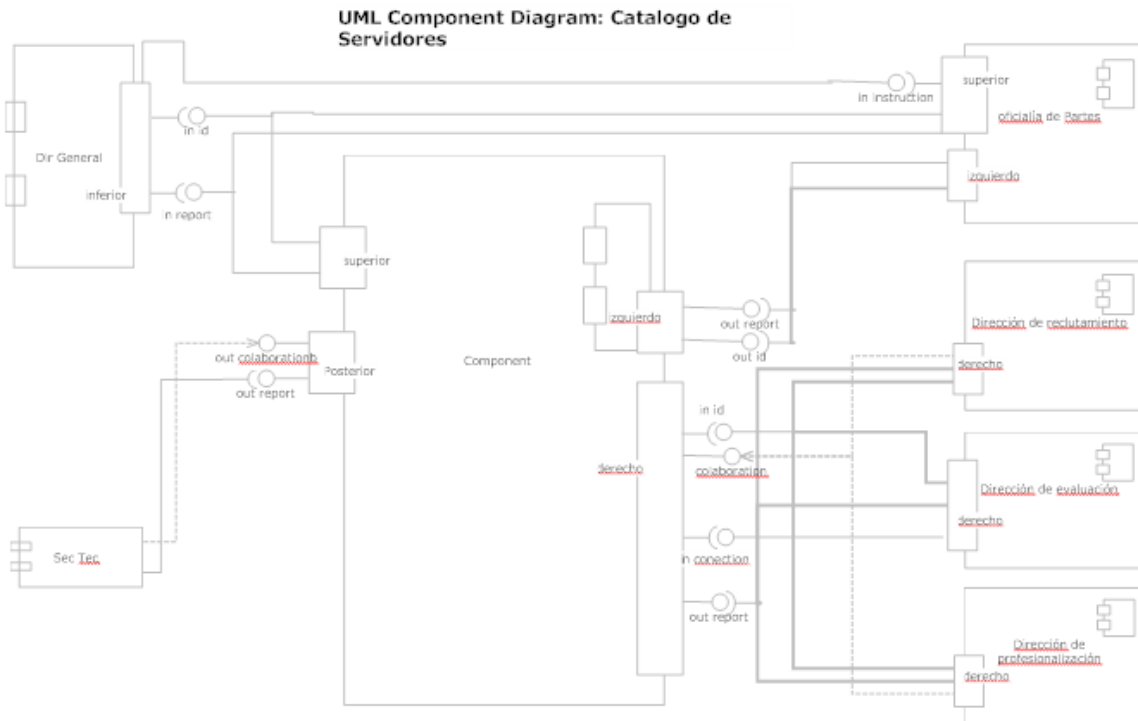


***Cubograma de la aplicación***

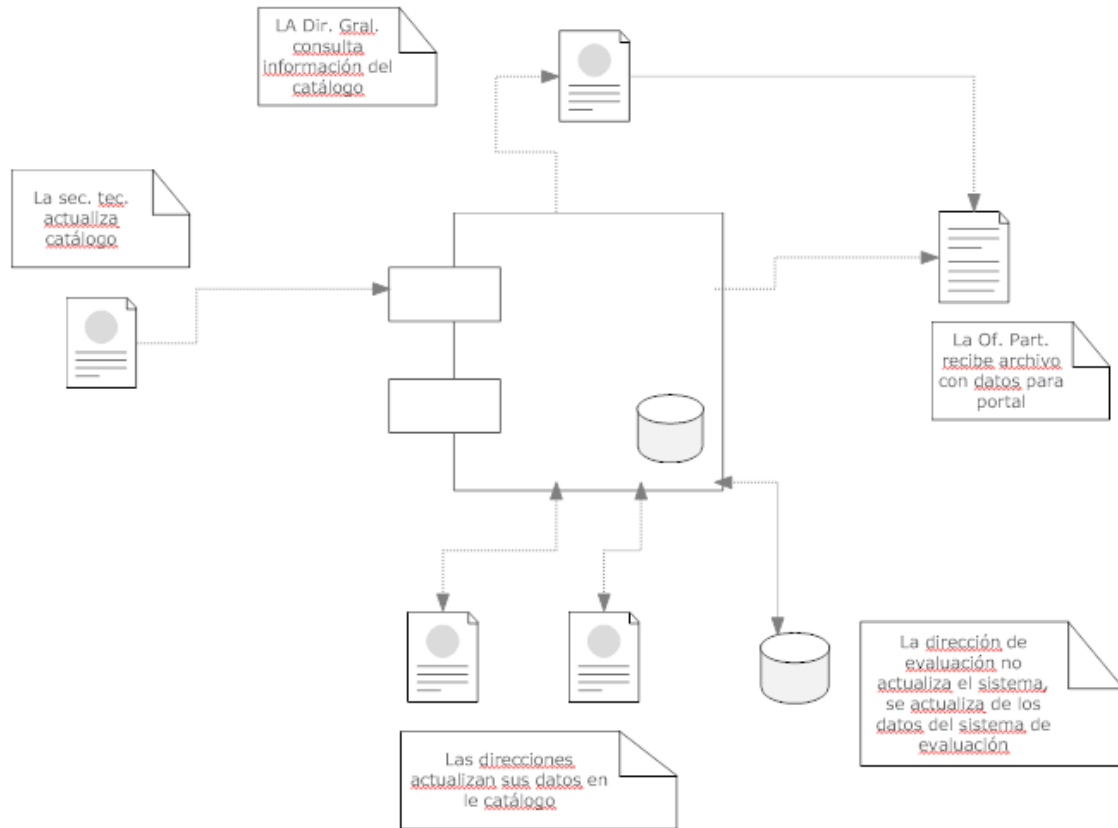




**Paso 6. Diagrama de componentes.**







**Descripción de la aplicación**



## CONCLUSIONES

El cubograma es una forma sencilla de hacer una representación de un sistema, que puede ser fácilmente entendible por el cliente y que fácilmente puede ser transformada en un diagrama de componentes que sirvan como una arquitectura de alto nivel para un sistema.

El cubograma es una propuesta de diseño de una arquitectura que pueda ser entendida por todos los involucrados en el proyecto, que pueda ser elaborada en las fases tempranas de desarrollo, e incluso pueda ser utilizada en la fase de requerimientos para conceptualizar de manera gráfica un sistema.

Esta propuesta va ser apenas utilizada en un proyecto real, se va a desarrollar una propuesta de arquitectura utilizando el cubograma como base.

Cabe mencionar que el cubograma como tal, debe ser utilizado como un enlace entre requerimientos y diseño, y como complemento en todo el proceso de desarrollo, para que los involucrados en las diferentes fases de desarrollo, puedan tener una perspectiva de todo el proyecto.

El producto resultante del cubograma permite la modificación del sistema, ya que el procesamiento se hace dentro del componente y solo exporta o recibe información, y en caso de ser necesaria una modificación se realiza dentro del propio componente sin afectar a los demás componentes. En caso de añadir un nuevo componente se realizan las modificaciones internas y se configuran las interfaces necesarias para su comunicación.



## REFERENCIAS

- [1]. Anónimo, “**Organigrama**”, Wikipedia agosto 2008, <http://es.wikipedia.org/wiki/Organigram>.
- [2]. Clements, Paul C., Northrop, Linda M., “**Software Architecture: An Executive Overview**”, Technical Report Software Engineering Institute, Carnegie Mellon University, February 1996.
- [3]. Garlan David, “**Software Architecture: a Roadmap**”, School of Computer Science, Carnegie Mellon University.
- [4]. Booch Grady, “**The irrelevance of Architecture**”, IEEE Software, may/june 2007
- [5]- Erikson, Hans-Erik., Penker, Magnus. “**UML Business Patterns at Work**”, Wiley Computer Publishing, 2000.
- [6]. Kruchten, et all., “**The past, present, and future of software architecture**”, IEEE Software, March/April 2006.
- [7]. Harmon, Paul., “**Developing an enterprise architecture**”, Bussines Process Trends, January 2003.
- [8]. Alexander, Ian., “**Building what stakeholders desire**”, IEEE Software, March/April 2007.
- [10]. OECD e-Government Studies, “**The e-Government Imperative**”, OECD 2003.
- [11]. Ivers, James et all. “**Documenting Component and Connector Views with UML 2.0**”, Technical Report Software Engineering Institute, Carnegie Mellon University, Abril 2004.
- [12]. Bass, Clements, Kazman , “**Software Architecture in Practice (2nd edition)**”; Addison-Wesley 2003.
- [13]. ANSI/IEEE Std 1471-2000, “**Recommended Practice for Architectural Description of Software-Intensive Systems**”.
- [14]. Thompson, Ivan, “**Tipos de Organigrama**”. Promo negocios 2006, <http://www.promonegocios.net/organigramas/tipos-de-organigramas.html>

