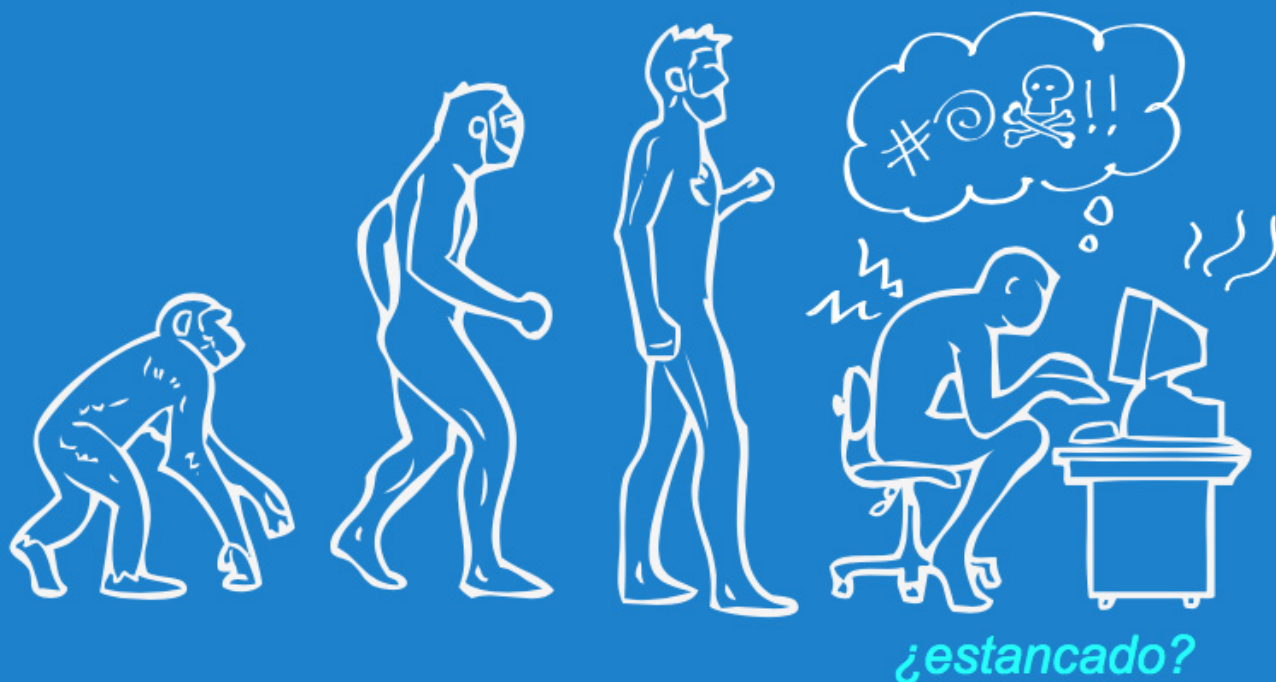




Guía básica para la
**Modernización
de aplicaciones**

Índice

1. INTRODUCCIÓN	2
2. METODOLOGÍA	6
3. HERRAMIENTAS	16
4. RESULTADOS	22
5. ANEXOS	28

1. INTRODUCCIÓN



1.1. Introducción

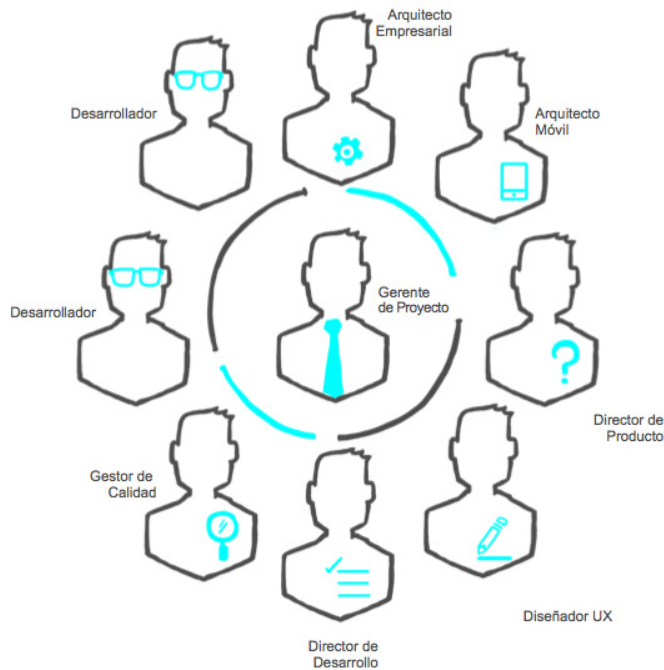
Los sitios web, web móvil y aplicaciones se vuelven cada vez más y más complejos en la medida en que la industria, la tecnología, las herramientas y las metodologías de desarrollo avanzan. Lo que solían ser medios rígidos y estáticos se ha transformado en espacios nuevos de interacción donde los usuarios tienen nuevas experiencias.

Sin importar que tanto ha cambiado la como los desarrolladores conciben aplicaciones y sin importar el trabajo de fondo que se le haya invertido a una iniciativa de desarrollo, al final el éxito depende de la percepción de un usuario que se pregunta. ¿Esta aplicación me genera algún valor? ¿Es útil? ¿Me agrada?

Responda a las necesidades de su negocio, lleve a su empresa a los más altos niveles de competitividad y saque el máximo provecho a sus productos Progress con la modernización de sus aplicaciones.

¿Qué es la Modernización de Aplicaciones?

Es el proceso en el cual se toma una aplicación legada hecha en 4GL/ABL Progress, y se cambia a nuevas tecnologías que permitan la convivencia con tecnologías legadas, siempre y cuando se genere valor y una experiencia positiva para el usuario.



1.2. Equipo

Muchas organizaciones han tercerizado los servicios de desarrollo lo cual trae ventajas y desventajas. Estas organizaciones suelen tener guías y políticas para sus comunicaciones, que no existen necesariamente para el desarrollo de software.

El grupo de trabajo necesario para la implementación de una metodología de modernización debería tener los siguientes roles:

Gerente de Proyecto: Es el líder del proyecto de modernización de cara al cliente, a los ingenieros y el equipo de trabajo.

Arquitecto de Aplicación: Define la estructura del proceso de modernización. Interactúa con el director de desarrollo, diseñadores y gestores de calidad.

Director de Desarrollo: Coordina el equipo de desarrolladores de las distintas capas.

Diseñador de la Experiencia de Usuario (UX): Es la persona encargada de definir los estándares de la interfaz gráfica para asegurar la usabilidad, utilidad y eficiencia de cara al usuario final.

Director de Producto: Define los casos de uso y la funcionalidad de las aplicaciones.

Conocimientos requeridos para el proceso de modernización :

- Conocimiento de programación ABL
- Conocimiento de programación orientada a objetos
- Conocer el objetivo de cada uno los componentes OERA [Arquitectura de Referencia Open Edge] o en su defecto los objetivos de SOA [Arquitectura Orientada a Servicios]
- Conocimiento del negocio.
- Conocimiento en desarrollo JavaScript
- Conocimiento HTML
- Conocimiento en análisis y modelado de flujos de trabajo.
- Conocimiento en análisis y modelado de reglas de negocio.

1.3. Bases para una Modernización Exitosa

Preservar el Core

Las aplicaciones empresariales suelen tener un desarrollo minucioso de su capa CORE, fruto de años de implementación, pruebas y correcciones, por lo cual el aprovechamiento de esta capa es fundamental en nuestra ruta de modernización.

Despliegue Híbrido

El proceso de modernización se hace a partir de entregas parciales que son desplegadas y funcionan de manera simultánea con la aplicación existente, esto permite mantener el servicio al usuario durante el proceso.

Desarrollo Ágil

Contar con una herramienta de rápido desarrollo que ayude al equipo de trabajo técnico a entregar aplicaciones en menor tiempo. Esta herramienta, complementada con el uso de metodologías ágiles, hace más eficiente el trabajo, baja los tiempos de desarrollo y permite el cumplimiento de los tiempos de entrega.

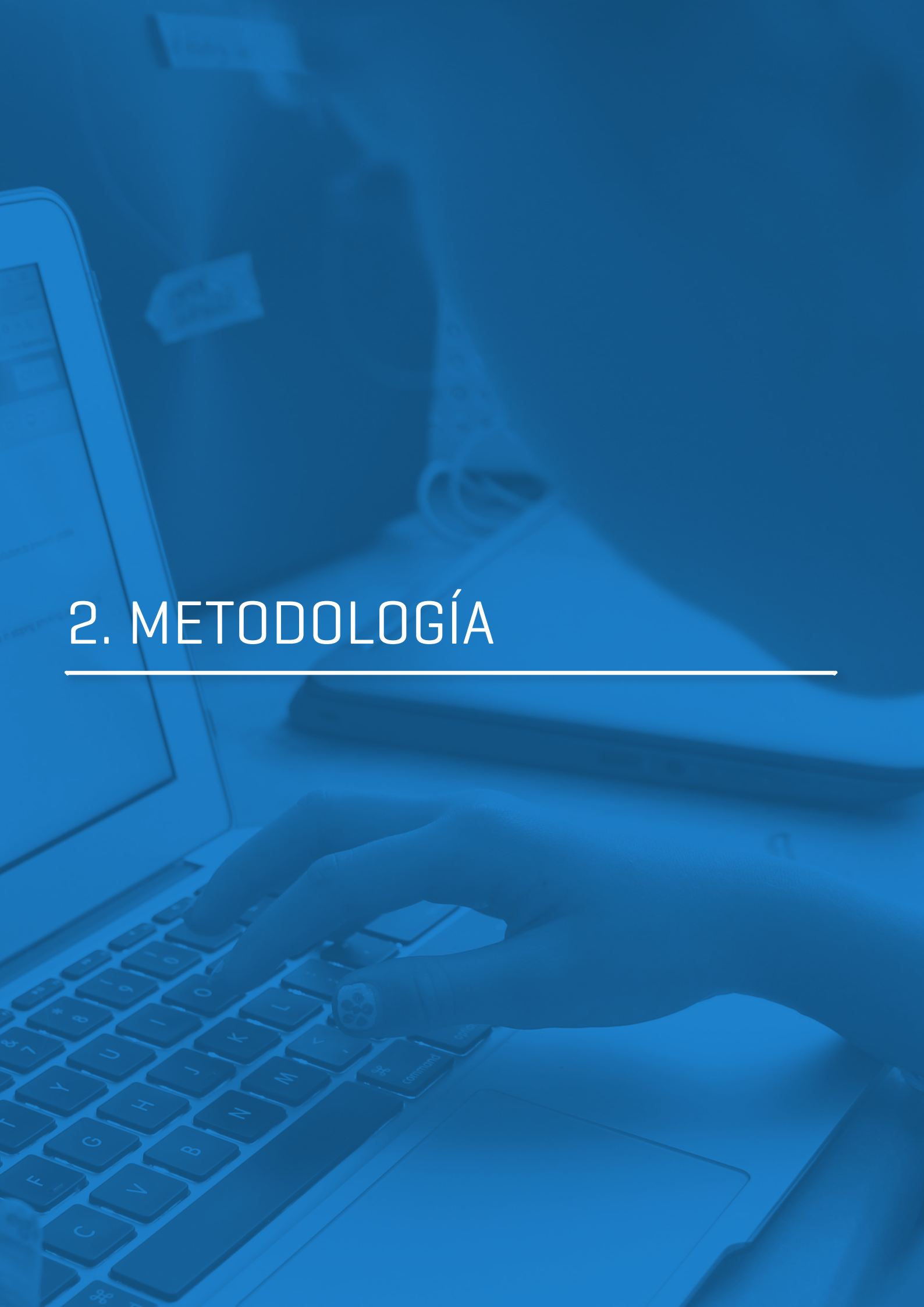
Diseño para el Futuro

Desarrollar aplicaciones pensadas para el futuro haciendo énfasis en la posibilidad de evolucionar y adaptarse rápidamente a los cambios del mercado y las necesidades de la empresa.

Experiencia de Usuario

El éxito de una aplicación depende de que en la práctica el usuario la encuentre útil, agradable y fácil de usar. En la metodología de Quantum traemos a las aplicaciones empresariales todos los adelantos propuestos por la industria en este tema, tomando como referencia las aplicaciones comerciales líderes en el mercado.





2. METODOLOGÍA

2.1. Arquitectura de desarrollo

La arquitectura técnica que se utiliza para el desarrollo de la modernización se basa en OERA (Arquitectura de referencia de OpenEdge), la cual permite una alta modularidad y escalabilidad. Finalmente se construye una arquitectura orientada a servicios [SOA] que va a permitir interactuar con cualquier interfaz de usuario o sistema. El equipo de Desarrollo de Quantum conformado por Expertos Progress, propone una arquitectura con las siguientes capas:

Presentación: Es la capa que ve el usuario final. Esta capa define la interfaz gráfica y proporciona características amigables para el usuario final.

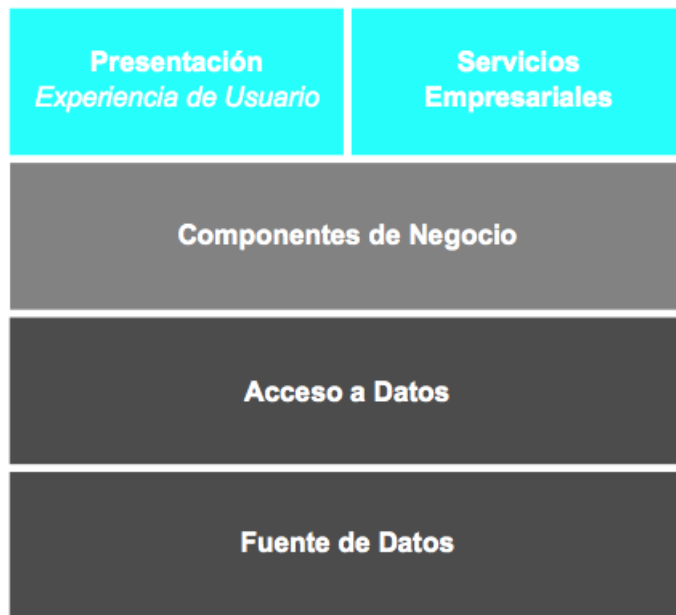
Componentes del negocio: Es la encargada de realizar todo el flujo y lógica del negocio que se requiere. Es donde residen los programas que expresan la lógica propia del negocio. Está compuesta por: reglas de negocio, Interfaces de servicio, Entidades, tareas y flujos de trabajo.

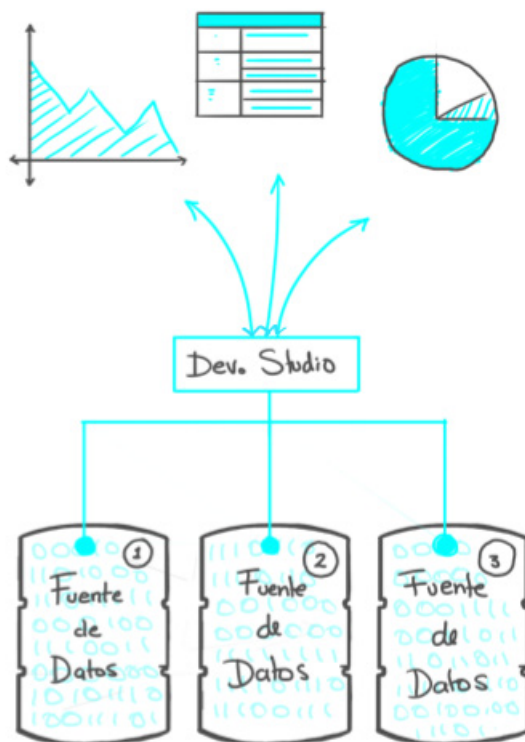
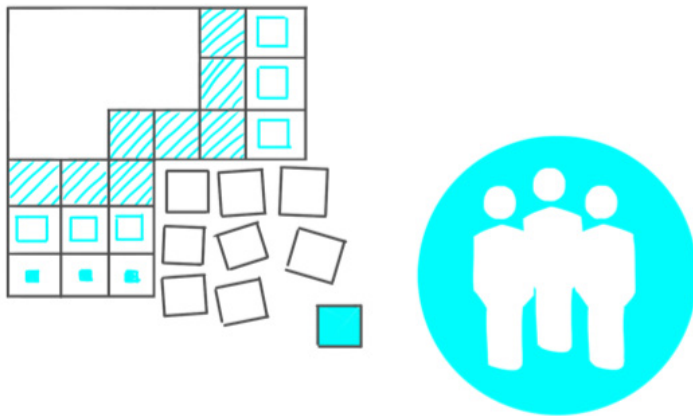
Acceso a Datos: Contiene los componentes que se encargan de gestionar el acceso a la fuente de datos.

Fuente de Datos: Son los motores de bases de datos, archivos, archivos XML, etc. en donde reside la información del negocio.

Infraestructura común: Define lo que es de uso común entre las capas.

Las capas de componentes del negocio y acceso a datos se denominan CORE para el desarrollo de la presente guía, la funcionalidad de estas capas será desplegada en el servidor de aplicaciones PAS. La capa de Presentación tiene dos partes: La primera comprende la lógica de eventos para lo cual estamos usando vue.js, kendo.ui, jquery y en la parte visual bootstrap.js que nos sirve para que las paginas sean responsive, por otra parte, para el despliegue de esta capa utilizamos Node.js, con express.js para el manejo de las rutas de la aplicación y jwt para el manejo de token.





2.2. Capa CORE

Requisitos

Analizar la aplicación, dividirla en módulos a modernizar, se sugiere que los nuevos requerimientos de la aplicación se desarrollen bajo los parámetros de modernización. [\[ANEXO I.\]](#)

Definir el equipo de trabajo que estará involucrado en el proceso de modernización tanto de parte de Quantum como del lado de la organización.

Migrar la base de datos y la aplicación a la última versión disponible de PROGRESS.

2.2.1. Desarrollo de la Capa de Acceso a Datos

La capa de Acceso a datos contiene los componentes y clases [.cls] de conexión a los datos. Es la única capa que accede de manera directa a los repositorios, y son utilizados desde la capa de lógica de negocio.

Para el desarrollo de esta capa se utiliza Prodataset, lo cual permite reutilizar el código en cada una de las tareas.

Es lo que se conoce típicamente como un CRUD [Creación, Lectura, Actualización y Borrado]. Para su modernización Progress Software ofrece Wizards en el Developer Studio que agiliza la generación de las clases y los componentes.

2.2.2. Desarrollo de la Capa de Componentes del Negocio

La capa de componentes de negocio requiere de programación orientada a Objetos, tablas temporales y Prodatasets. Se compone de interfaces de servicio, entidades de negocio, tareas de negocio y flujos de trabajo.

Definición de Servicios:

Los servicios incluyen los parámetros de entrada y salida definidos para el servicio a desarrollar. [\[ANEXO II.\]](#)

Se deben definir 2 tipos de servicio: uno para Consultar y otro que Crear, Actualizar y Eliminar.

Definición de las Entidades de Negocio:

Las entidades de negocio son una representación en memoria de una entidad de la Base de Datos, se debe definir en un dataset o en una tabla temporal. [\[ANEXO III\]](#)

Definición de las entidades de tarea:

Las entidades de tarea permiten un puente entre el flujo de trabajo y la capa de acceso a datos. se debe componer por una clase .cls y la definición de sus métodos deben responder a las necesidades del flujo de trabajo. [\[ANEXO IV\]](#)

Definición del Workflow:

El workflow debe contener el código necesario para responder a las necesidades del negocio definidas para el servicio. dentro este tenemos la definición de los siguiente items:

- **Interfaces de Servicio:**

Son las encargadas de exponer hacia la capa de presentación los parámetros definidos en el servicio. Son muy livianas en su programación ABL [Advance Business Language] con extensión .p, no realizan lógica de negocio alguna, sino validación de parámetros (tamaños y tipos de datos), recibo y entrega de datos en los diferentes formatos requeridos.

- **Entidades de Negocio [Entity]:**

Representan una entidad del modelo de datos. Para esto se debe crear un "Business logic component" por cada entidad de negocio, o según lo que se requiera por el servicio definido, este debe estar compuesto por una o más tablas temporales.

- **Tareas de Negocio [Task Entity]:**

Se encargan de gestionar la conexión con la capa de acceso a Datos y deben proveer al flujo del negocio (workflow) la información que sea requerida.

- **Flujos de Negocio [Workflow]:**

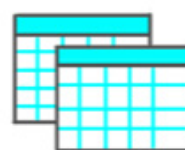
Se encargan de definir toda la lógica de negocio, y es donde puede persistir la información de la Entidad de Negocio.

Serv.	Entrada/Salida	Datos
Consulta Cliente	Entrada	Nombre
		Cédula
		Fecha
	Salida	Productos
		Saldos
		Ventas



Durante este proceso se emplean herramientas de trabajo colaborativo como Google drive para asegurar el trabajo coordinado del equipo.

[ANEXO V.]



Producto

	Categoría
	Cantidad
	Empaque
	Nombre
	Código

2.3. Capa Intermedia

- Orquestador

En la capa intermedia de la aplicación se encuentra el servidor de aplicaciones de progress [PAS], este es el servidor de aplicaciones web principal, basado en Apache Tomcat, que es la base de los servidores de aplicaciones en OpenEdge. Pacific Application Server para OpenEdge [PAS para OpenEdge] es un servidor de aplicaciones específicamente diseñado para soportar aplicaciones OpenEdge incluyendo aplicaciones WebSpeed.

PAS para OpenEdge está disponible como dos herramientas separadas:

- **The Pacific Production Application Server for OpenEdge** — configurado como un servidor web seguro para la implementación de la aplicaciones OpenEdge.
- **The Pacific Development Application Server for OpenEdge** — configurado como un servidor web para desarrollar y probar aplicaciones OpenEdge.

Nota: PAS para OpenEdge y OpenEdge AppServer son dos servidores de aplicaciones únicos y separados en OpenEdge. Ambos admiten el desarrollo y la implementación de aplicaciones ABL, pero son completamente diferentes en sus arquitecturas.

Progress Developer Studio [PDS] para OpenEdge soporta desarrollo de aplicaciones, pruebas, depuración e implementación, que incluye un tipo de proyecto y un servidor de desarrollo para PAS para OpenEdge.

¿Qué orquesta?

Dentro de la aplicación es el encargado de gestionar todo el flujo de trabajo desde la conexión a las bases de datos, conexión a servicios de terceros (alfresco, bpm, corticon, mongo, entre otros), bien sea por medio de API's (REST o SOAP), como a través del propio sistema operativo anfitrión.

2.3.1. Reglas de Negocio

Corticon proporciona una metodología simple pero potente para el modelado de las reglas de negocios. La metodología consiste en los siguientes pasos

- **Modelado:** Se diseña el vocabulario o términos conceptuales de la aplicación.
- **Análisis:** Se construyen las reglas de acuerdo al Core del negocio.
- **Pruebas:** Se realizan las pruebas al modelo.
- **Despliegue:** Se despliega el Servicio de Decisión para que sea consumido por cualquier método de integración.

Dentro del proceso de modernización existen 4 tipos de integración:

- **Asignación directa de objetos:**
Se realiza incorporando una biblioteca con una API definida dentro de en una aplicación tal como lo haría con cualquier biblioteca de terceros. Para Java se incluye archivos JAR y para .NET se proporcionan conjuntos DLL.

Este modelo de implementación permite consumir un servicio de decisión con los objetos de datos de su aplicación a través del uso y configuración de un adaptador nativo. [BPM]. También puede consumirse con instrucciones ABL dentro de nuestra lógica de aplicación.

- **SOAP Web Services:**
Este es el modelo de implementación más común utilizado por los clientes de Corticon. Permite implementaciones que aprovechan la arquitectura orientada a servicios basada en estándares conocidos.

Está basado en la entrega de archivos XML o JSON como resultado o consumo del servicio de decisión.

- **REST Web Services:**
La API REST para el consumo de servicios de decisión es bastante nueva para Corticon, pero podría superar rápidamente a la interfaz SOAP. Comparte la mayoría de los beneficios del SOAP debido a la naturaleza desacoplada de su arquitectura.

La forma de consumo de la API REST con JSON, es bastante sencilla y sigue las convenciones típicas de REST API.

El modelo de objetos JSON, está documentado en la sección de ejemplos de solicitud y respuesta en la documentación de Corticon.

- **Servicios Java J2EE:**
Corticon también está disponible como un bean de sesión sin estado J2EE (EJB) que le permite ser parte de una aplicación que aprovecha la pila J2EE completa.

Este modelo le permite invocar servicios de decisión a través de Java Messaging Service [JMS] o Remote Method Invocation [RMI].

Corticon proporciona mucha flexibilidad para incorporar capacidades y flexibilidad de reglas en su aplicación. El modelo de integración que debe utilizarse y sus ventajas depende de la estructura de su aplicación.

[ANEXO X*]

2.3.2. Gestión de documentos

La gestión de documentos se soporta con el uso del motor de Alfresco ECM. Alfresco ofrece la capacidad de ser columna vertebral de contenido para cualquier organización con la necesidad de capturar, controlar y administrar su contenido. Dado que soporta estándares abiertos hace que sea fácil de integrar con otras aplicaciones comerciales. La plataforma abierta evita el bloqueo de proveedor.

Toda la funcionalidad de administración de contenido central es proporcionada por el nivel de servicio. Los servicios principales incluyen:

- **Metadatos:** definidos por tipos de documentos y aspectos. Una gama de conjuntos de metadatos predeterminados están disponibles de forma inmediata, pero generalmente se agrega un modelo de metadatos personalizado.
- **Versiones:** de cualquier tipo de contenido, no hay restricciones basadas en el tipo de archivo. Se proporciona un historial de versión completo e incluso los cambios de metadatos son versionados.
- **Transformaciones:** de los tipos de contenido más utilizados, por ejemplo, MS Office a PDF. El servicio de transformación también se usa para generar vistas previas de contenido y miniaturas.
- **Búsqueda e indexación:** hace posible que los usuarios encuentren contenido. Tanto los metadatos como el contenido del archivo se pueden buscar.
- **Permisos:** controla lo que un usuario puede hacer. Los permisos se pueden establecer para un usuario o grupo en carpetas o documentos individuales.
- **Audit trails:** permite registrar cualquier acción en el sistema. La auditoría es configurable y por artículo o acción en el repositorio.

2.3.3. Gestión de Procesos

En un proceso de modernización se debe tener en cuenta cuando una aplicación es transaccional o realmente hace parte de un flujo de proceso. Una manera sencilla para identificar si algunas transacciones hacen parte de procesos es responder unas simples preguntas:

- ¿Existe más de una persona envuelta en la ejecución apropiada de la transacción?
- ¿Existe intercambio de información y de actividades entre las personas?
- ¿Se necesita tener un seguimiento de las actividades realizadas. Cuando se hicieron, cuanto demoraron, donde existe el cuello de botella...?
- ¿Si las respuestas a estas preguntas son positivas, las transacciones se deberían organizar dentro de un gestor de procesos?

La metodología Integrity Procesos se organiza en tres etapas: descubrimiento, implementación, monitoreo.

- El insumo o entrada es la selección del proceso objetivo.
- En la etapa de descubrimiento se revelan los elementos del proceso y el entregable es un modelo descriptivo que se trabaja en el modelador de procesos ([ver OpenEdge Business Process Modeler](#)).

- » Unidades organizacionales, actores, roles, documentos, productos, datos, aplicaciones, servicios, actividades, eventos, reglas, flujos, reportes e indicadores, son elementos del proceso.
- » Se hace énfasis en detallar las entradas, las salidas, el qué, el cómo y el quién de las actividades del proceso a través de un documento plantilla estándar que recopila estos datos como información preliminar del proceso.
- » Se logra un modelo inicial que captura el estado actual del proceso.
- » Se realiza análisis sobre el modelo para identificar posibles problemas en el proceso.
- » En caso de aplicar mejoras, se obtiene entonces un modelo final del proceso conforme a los requisitos.
- En la etapa de implementación el modelo descriptivo se transforma en un modelo ejecutable del proceso.
 - » Entendimiento del modelo de proceso, el modelo de datos y la experiencia de usuario [formularios para tareas de usuario].
 - » Se ajusta el modelo descriptivo para tener un modelo ejecutable utilizando la herramienta Progress Developer Studio for OpenEdge.
 - » Desarrollo de interfaces para la integración de la capa de presentación con el modelo ejecutable haciendo uso del API OE-BPM.
- En la etapa de monitoreo se hace un chequeo continuo para medir el comportamiento y desempeño del proceso.

[ANEXO X, XI, XII]

2.4. Capa de Presentación

2.4.1. Modernización Orientada a Aplicaciones

Las aplicaciones web se despliegan en Node.js, esto hace que la capa de presentación en el caso de las páginas web sea escalable para reutilizar el código tanto en navegadores web como en dispositivos móviles u otro dispositivo, ésta capa tiene una arquitectura diferente, una arquitectura orientada a eventos. La estructura o componentes en los que está construida se divide en dos partes: una del lado del servidor y la otra del lado del cliente.

Capa del lado del servidor, como su nombre lo indica ésta capa hace de servidor de aplicaciones, ya que está soportada en Node.js, según los propios creadores, Node.js no es un servidor de aplicaciones pero tiene un ecosistema de paquetes para disponibilizarlo, es mucho más rápido, liviano y eficiente, ya que manejan una arquitectura orientada a eventos, adicional a esto soporta grandes cantidades de usuarios evitando que las aplicaciones que se despliegan tengan menos probabilidades de un ataque DDOS [ataque de denegación de servicio].

Con uno de los paquetes de npm que es ecosistema más grande de paquetes, se pueden estructurar los niveles de seguridad de la página a través del paquete jsonwebtoken.

Capa del lado del cliente o capa pública, esta es la capa que ve el usuario en el navegador web, la cual está construida con diferentes componentes y herramientas entre los cuales tenemos a vue.js como framework a nivel de lógica de presentación, bootstrap y kendo. UI como framework de la vista y otra como jquery.js y ajax que nos sirve para conectarnos con la capa Core.



Vue.js como se comentó anteriormente es el framework que se utiliza para construir la lógica de la capa de presentación, este framework esta presente tanto en los documentos html css y js, y facilita el desarrollo de componentes como arboles, grillas y algunas vistas que son especiales para cada aplicación modernizada que se pueden replicar como componentes de vue.js.



Bootstrap, permite que todas las aplicaciones sean responsive, esto quiere decir que mantengan una buena visualización si se despliega de cualquier dispositivo sin importar el tamaño.



Kendo UI®

Kendo UI, se utiliza para visualizar listas Grillas con sus respectivos widgets que son bastantes y facilitan la búsqueda de grandes volúmenes de información.



jQuery.js, facilita la conexión con los servicios de la capa CORE de la aplicación a través de los servicios REST disponibilizados por el PAS.

2.4.2. Modernización Orientada a

la Gestión De Procesos De Negocio

La experiencia del usuario es alrededor de tres vistas o perspectivas: procesos, micro-flujos, tareas ad hoc [ver punto 3.1.4.].

Todas las tareas son organizadas en una bandeja central que ofrece de manera amigable la información preliminar o atributos de la tarea clave: el nombre de la tarea, una breve descripción, la fecha de activación, la fecha de vencimiento, el estado. Opciones de ordenamiento son disponibles por cada atributo.

2.4.3. Modernización Orientada a Aplicaciones Móviles

Tipos de Aplicaciones Móviles.

Existen básicamente 3 tipos. Nativa, Híbrida y Web Móvil. De acuerdo a la necesidad de la empresa se debe usar una de estas:

Nativa

Si la aplicación hace uso intensivo de los periféricos del celular[tales como el acelerómetro, NFC, identificador de huella y otros] o requiere un alto consumo de procesamiento, como los video juegos, en el móvil directamente la mejor solución es escribir una aplicación en el sistema operacional del dispositivo, o nativa. Si la aplicación accesa algunos periféricos estandar como la cámara de fotografía o el GPS unicamente lo ideal es construir una solución web móvil. HTML5 ya tiene la definición y manejo de estos periféricos.

Esta es la aproximación más usada en Quantum porque permite tener un único desarrollo móvil y web. La misma aplicación web que funciona en navegadores convencionales podría funcionar el celular. Lo único que se exige es que la aplicación que se desarrolle para web cumpla con ser responsiva y adaptativa a la vez. Esto quiere decir que se adapte de acuerdo al dispositivo que la llama, pueda restringir funcionalidad o cambiar la navegabilidad si fuera necesario.

Híbrida

Es una mezcla de las dos anteriores. Son aplicaciones que tienen parte de su funcionalidad nativa y parte web móvil. Aquellas aplicaciones que son web en su esencia, pero hacen uso frecuente de los periféricos del celular podrían ser las ideales para un desarrollo híbrido. Un ejemplo de esto es una aplicación que haga la autenticación en el dispositivo, pero el procesamiento en el back end.

Si la necesidad es de construir una aplicación nativa o híbrida en Quantum usamos Native Script. Esto va a permitir hacer un solo desarrollo independiente de plataforma[IOS, Android o windows].

En la implementación se tienen cuatro grandes fases:



Construcción de la interfaz de usuario: se definen un mockup con todas las pantallas y su contenido; esto se hace para tener un primer alcance de lo que puede llegar a ser la aplicación móvil.

Luego se construyen los archivos xml para la estructura de la interfaz y se estiliza usando css para dar vida a las pantallas modeladas en el mockup.

Adición de eventos y programación JavaScript: los componentes móviles manejan eventos (presionar un botón, hacer zoom, etc.) a los cuales se les deben definir propiedades y acciones programadas en JavaScript o TypeScript. Estos componentes pueden ser Kendo UI u otros.

Conexión al lado CORE: las pantallas desarrolladas anteriormente se deben conectar a una "solución" del lado CORE, es decir a los servicios REST disponibles a través del AppServer y esto se resuelve a través de peticiones http (POST,GET,UPDATE y DELETE)

Pruebas y descarga de la aplicación: finalmente la aplicación se prueba en un simulador web, el cual permite corregir los errores de la aplicación. Finalmente se producen los archivos que se pueden descargar a dispositivos móviles Android, iPhone OS y Windows Phone.



2.5. Experiencia de Usuario

Las aplicaciones son herramientas hechas para un usuario final que le dan la oportunidad de generar, consultar e interactuar con la información. En este contexto existe un factor humano sumamente importante que se debe tener en cuenta para el diseño de una Experiencia de Usuario.

¿Quién es usuario? ¿Cuáles son sus capacidades, habilidades y motivaciones? ¿Cuáles son sus necesidades y como nos planteamos solucionarlas?

Entendiendo esto y el contexto en el que se hará uso de las herramientas se debe definir una jerarquía lógica de la información y sus niveles, un mapa de navegación eficiente y fácil de entender, el diseño de interfaces coherentes y consistentes que le permitan recorrer con seguridad la aplicación, entender su constitución, sus alcances y limitaciones y ejecutar procesos de manera intuitiva y eficiente.

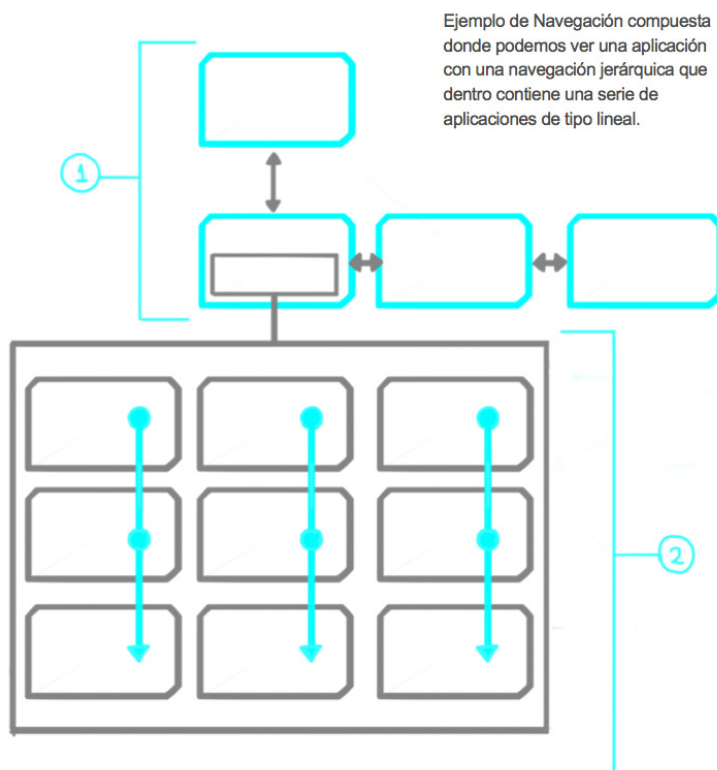
Todo este esfuerzo está enfocado en generar una Experiencia de Usuario positiva incidiendo en los aspectos prácticos, emocionales, experimentales y afectivos de la interacción del usuario con un sistema informático. El objetivo es controlar la manera en que el usuario percibe la herramienta en términos de utilidad, eficiencia y usabilidad.

A partir de la experiencia que nos ha permitido el trabajo en Quantum Data Systems, nuestro equipo de profesionales interdisciplinarios y altamente capacitados ha desarrollado unas pautas básicas para afrontar estos retos de Modernización de Aplicaciones en términos de la experiencia de usuario y conseguir la máxima satisfacción alcanzando los objetivos planteados por la empresa.

2.5.1. Mapa de Navegación

Un mapa de navegación es un esquema que define la estructura de la aplicación y representa a través de posición, flechas y colores las páginas y posibles direcciones en las cuales un usuario puede desplazarse a través de la aplicación. Algunos de los tipos de navegación más comunes son:

- **Lineal:** Usualmente implementada en procesos que implican una secuencia de acciones en un orden lógico preestablecido. Permite navegar únicamente de una pantalla hacia la siguiente o a la anterior.
- **Estrella:** La navegación en estrella consta de una pantalla central desde la cual se puede acceder a pantallas secundarias y regresar.
- **Jerárquico:** Es un tipo de navegación que distingue claramente la jerarquía de la información que presenta la aplicación. Este mapa es especial para el manejo de grandes cantidades de información.
- **Compuesto:** Usualmente el desarrollo o modernización



de aplicaciones hace necesaria la implementación de varios tipos de navegación. En estos casos se utiliza un mapa de navegación compuesto teniendo clara la importancia de establecer un criterio que defina en qué casos se aplica un determinado tipo de navegación.

2.5.2. Look & Feel

Para el diseño del look and feel en la modernización de aplicaciones es importante tener en cuenta la comunicación de la marca de la empresa dueña de la aplicación con el objetivo de hacer un diseño coherente con su discurso.

Esto hace que la aplicación sea más fácil de asimilar y entender por la familiaridad que tiene el usuario con estos aspectos formales de la comunicación y la relación que ha hecho de dichos aspectos con la marca.

Como resultado se deben establecer pautas al respecto de los colores a utilizar, los tipos de imágenes, las fuentes y sus tamaños, la implementación de íconos y la naturaleza formal de los mismos.

2.5.3. Diseño de Pantallas

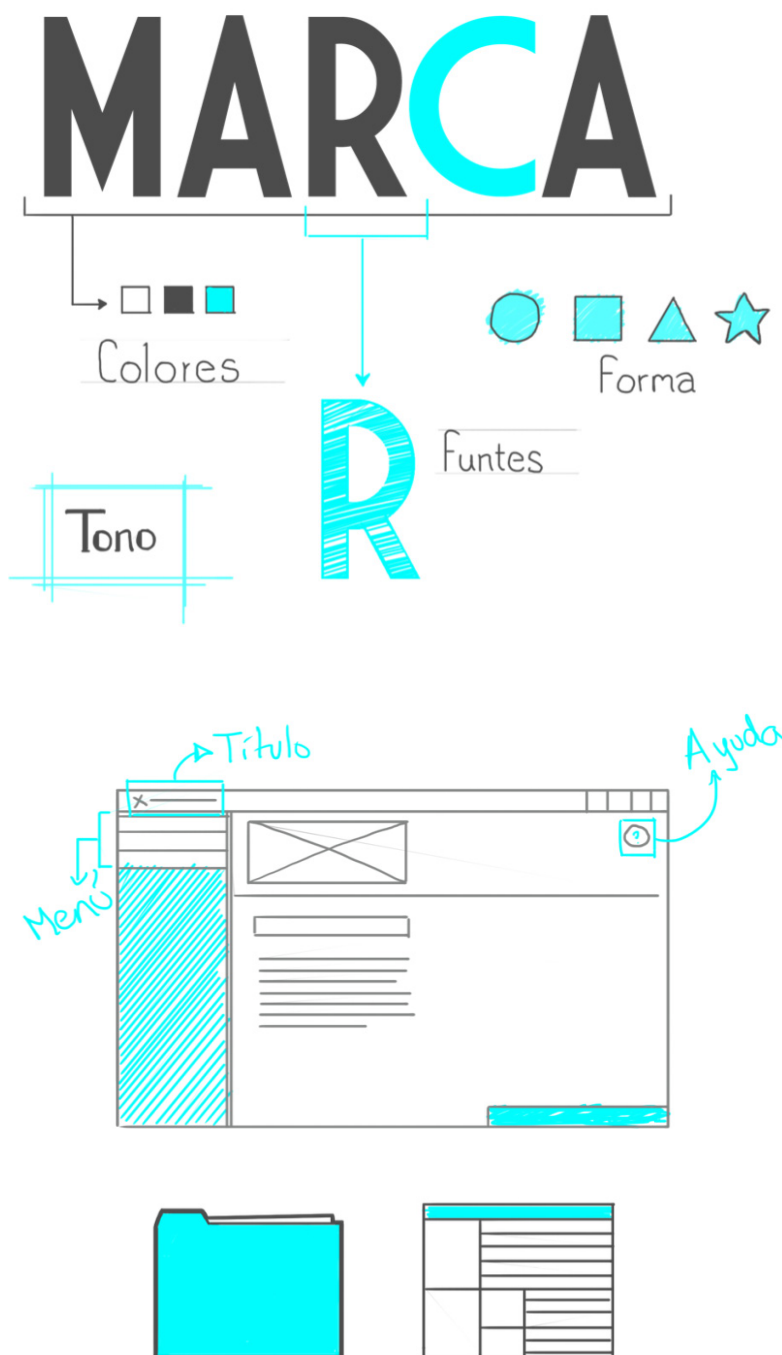
A partir del estudio de los procesos y el mapa de navegación se identifican y clasifican aquellos que contienen estructuras y funciones similares. Para cada uno de estos grupos se diseña un formato específico de pantalla. Los procesos que tienen una naturaleza única son clasificados como atípicos y se diseña una pantalla específica, única para este proceso. Para el diseño de estas pantallas es especialmente importante basarse en los estándares definidos en look and feel de la aplicación.

El diseño de estas pantallas se hace directamente sobre la herramienta Developer Studio o el Telerik AppBuilder [Móvil] en donde se define la estructura principal de sus pantallas y su navegación. El directorio de herramientas prediseñadas que ofrecen programas como Kendo UI hacen del diseño de pantallas algo dinámico, ágil y sencillo.

El Diseño de pantallas debe dar como resultado una lista clara y completa de las pantallas a desarrollar, acompañado con los formatos de pantalla hechos en OE BPM, listos para que los implementadores comiencen su labor.

2.5.4. Directorio Gráfico

La coherencia y la consistencia dentro de la interfaz gráfica de una aplicación es fundamental para lograr una Experiencia de Usuario Positiva. Por esta razón es importante, después de definir el Look and Feel y el diseño de las pantallas, estructurar un directorio gráfico en donde se definan con claridad las pautas para el desarrollo de imágenes, íconos, botones, colores, fuentes etc.



Este directorio dará las pautas formales de todos los elementos gráficos que conformarán la interfaz para asegurarse que en la aplicación se mantengan coherentes y consistentes entre ellos mismos y con el lenguaje tácito que ya existe para estándares de navegación.

Esto permitirá apropiarse de simbolismos prácticos que resumen la función de un botón a partir de un ícono que el usuario podrá entender intuitivamente. La implementación de códigos consistentes de color permiten una comunicación muy eficiente y un entendimiento intuitivo por parte del usuario.



3. HERRAMIENTAS

3.1. Herramientas Quantum

3.1.1. Plantilla

Se concibió como una herramienta front-end y back end, que permite reducir el esfuerzo para nuevos despliegues, utilizando las mejores prácticas en cuanto a materia de desarrollo. Escrita utilizando ABL para la capa core y JavaScript para la capa de presentación.

[ANEXO VI]

Front End

Por medio Javascript y JQuery se interpreta un JSON que trae la información de las grillas y formularios que se muestran al usuario. Esto permite que los textos y todos los elementos de la página sean ajustados a las necesidades de cada cliente. Hacemos uso de Kendo UI para lograr una experiencia amable de cara al cliente garantizando compatibilidad con la mayoría de los navegadores actuales.

Maestro Detalle

Esta herramienta permite, previa una configuración, mostrar y modificar los datos presentados por un servicio REST preconfigurado en forma de grilla, tanto la cabecera como el detalle. esta información se puede presentar como una tabla única ó como un maestro-detalle.

Back End

Es una herramienta desarrollada utilizando código progress [ABL] para generar aplicaciones [ABL] utilizando el modelo OERA como referencia, esta herramienta trata cada tabla física como una entidad, generando la definición de la tabla, Task, Workflow, Procedimientos de invocación y clase para ser expuesta como servicio REST, es capaz de generar un CRUD completo sin ninguna intervención, este CRUD no incluye ninguna lógica de negocio ya que se supone es diferente para cada uno.

3.1.2. Herramienta Administrativa

Brinda la posibilidad de realizar una serie de configuraciones sobre el portal, se concibe como una herramienta global implícita dentro del producto, la cantidad de modificaciones u opciones que permite es determinado por la disponibilidad de los módulos de Integrity, como lo son: Procesos, Reportes, Bloques e Integrity Lite.

Proceso de Autenticación

A nivel de back-end, el proceso de autenticación utiliza la validación de la base de datos Progress, primero autenticándose contra ella por medio del PAS, esta autenticación genera una sesión que es convertida a un formato de texto en base64, el cual es transferido al cliente a través el servicio REST de logueo, cualquier modificación en el mismo lo hace no parseable de vuelta por el appserver. Cuando se solicita cualquier información al appserver luego del logueo debe suministrarse esta sesión, la cual almacena todos los datos del usuario contra la base de datos, cabe destacar que el resultado de decodificar dicho base 64 es una data ilegible, ya que corresponde a un token de seguridad ABL. Además todo el canal de transporte de datos se asegura utilizando un certificado SSL, esto aplicado tanto para el back-end como para el front-end.

En la capa front-end, se hace uso de Json Web Token [JWT], para generar tokens aleatorios al momento de realizar el logueo, cualquier alteración en este token web conlleva a un direccionamiento inmediato a la pantalla de login y salida segura del sistema, así también la anulación del token ABL.

Manejo de Roles/Perfiles y Funciones

Cada usuario del sistema se asocia a un código de rol, el cual tiene un pool de funciones asociadas, estas funciones especifican a cuáles pantallas web puede ingresar el usuario, si el rol o la función es modificada se refleja inmediatamente en el menú del usuario en cuestión. Es importante acotar que por seguridad la páginas no se ven afectadas por modificaciones realizadas en la URL, ya que los direccionamientos se hacen de manera interna, cualquier modificación conlleva a un direccionamiento inmediato a la pantalla de login y salida segura del sistema, así también la anulación del token ABL.

Calendarios Corporativos.

Con el uso de la herramienta Open Edge BPM se cuenta con el manejo del calendario corporativo, lo cual permite que las tareas asignadas al usuario se hagan dentro de un horario laboral, y el control de la duración de estas tareas será registrado por el tiempo que transcurre en el calendario definido. El calendario permite la definición de días laborales, horario y hora de almuerzo.



3.1.3. Reportes

Reportes Quantum es una herramienta iconográfica, intuitiva, fácil de usar y de usuario final, que le permite a los empleados de su organización con conocimiento en su área hacer reportes de forma rápida sin necesidad de que intervengan los ingenieros de desarrollo.

Tengamos en cuenta que del software desarrollado en una empresa el 70 % son reportes y consultas, con esta herramienta se podrá minimizar este trabajo, además liberar a los desarrolladores para que concentren sus esfuerzos en otras tareas.

[ANEXO VII]

Características:

- ♦ **Usuario Final:** Permite crear reportes a la medida sin necesidad de involucrar al equipo de tecnología de su empresa.
- ♦ **Iconográfica:** Como sabemos es un lenguaje universal que hace que la herramienta sea más intuitiva.
- ♦ **Filtros:** El reporteador permite filtrar la información de acuerdo a lo deseado por el usuario y modificarlos en el momento que desee.
- ♦ **En tiempo real:** Esta propiedad es muy importante ya que nos permite tener la data hasta el momento de su ejecución.
- ♦ **Permite hacer gráficos:** Podemos utilizar diferentes gráficos como barras, torta, columnas, calendarios entre otros, para representar la información del reporte.
- ♦ **Seguridad:** un administrador puede hacer que cierta información no se vea para el usuario final.
- ♦ **Dashboard:** Poderosa característica que nos permite estar actualizados de la información de estos reportes al momento y analizar varios al mismo tiempo.

3.1.4. Tareas Ad Hoc

Con tareas ad hoc, los usuarios no tendrán que esperar más por la liberación de un proceso formal para poder registrar y administrar tareas. Estas tareas nacen de manera espontánea y en cualquier momento. Sus características más importantes son:

- Cero-programación
- No requiere apoyo en ningún momento de TI.
- No requiere conocimientos en modelado de procesos.
- La asignación de la tarea ad hoc es a sí mismo o un compañero de trabajo.
- En cualquier momento se puede reasignar la tarea ad hoc a un compañero de trabajo.
- Puede programarse la activación de la tarea ad hoc en una fecha y hora específica.
- Se configura la duración de la tarea ad hoc para indicar su vencimiento.
- A partir de una tarea ad hoc se puede establecer subtareas, lo que también se denomina en Integrity como tareas colaborativas.

[ANEXO VIII]

3.1.5. Micro-flujos

Con micro-flujos, los usuarios tienen la capacidad de crear, publicar y compartir flujos de trabajo muy sencillos basados en patrones de modelo predefinidos. Estos procesos no son misionales, pero si tienen gran importancia para la organización. Desafortunadamente la organización no los implementa por estar concentrada en los core. Algunos de estos ejemplos serían aprobaciones en los diferentes departamentos, manejos de caja menor, etc. El objetivo es que el usuario final sea capaz de definir, implementar y publicar estos flujos sin necesidad de apoyo por el departamento de tecnología.

[ANEXO IX]

- Soporta todo lo relacionado con tareas ad hoc
- El usuario sólo debe reconocer patrones de control del flujo y seleccionar el más acorde.
- Tres patrones básicos y recurrentes son soportados: lineal, condicional, ciclo.
- Soporte de definición de datos del proceso basado en tipos: texto, numérico, lógico, fecha, archivo, lista.
- capacidad de definir un número amplio de procesos administrativos

3.1.6. Proceso

Con procesos, se implementan flujos de trabajo complejos y hechos a la medida. Algunas de sus características más importantes son:

- Se aplica metodología especializada para el diseño, automatización y mantenimiento de procesos de negocio [ver metodología -> gestión de procesos].
- Se utiliza un lenguaje común para el modelo y notación del proceso: BPMN. Así, tanto en etapa de descubrimiento e implementación, como también en la ejecución del proceso y monitoreo, los analistas, los implementadores y los usuarios finales tienen el mismo entendimiento del modelo de proceso gracias a una única representación del mismo como uno o más diagramas BPMN [ver OpenEdge Business Process Modeler].
- Diseño de formularios a la medida para tareas de usuario, con soporte de controles web avanzados.
- Integración con diversas fuentes de datos para el soporte de tareas de servicio [ver Progress Developer Studio for OpenEdge].
- Soporta la definición, integración y despliegue de reglas del negocio.
- Se hace la integración con el motor de procesos BPM haciendo uso del API OE-BPM [ver BP Server].

[ANEXO X, XI, XII]



3.2. Herramientas Progress



Kinvey

es una plataforma de nube comercial para proporcionar Backend para dispositivos móviles, tabletas y aplicaciones web.

Kinvey le permite desarrollar aplicaciones rápidamente. El servidor de nube ofrece un SDK de cliente enriquecido, microservicios basados en funciones, integraciones empresariales rápidas y flexibles, capacidades robustas sin conexión, caché en la nube y seguridad.

OpenEdge-BPM

OE-BPM o Progress OpenEdge Business Process Server es una plataforma integral de gestión de procesos, que permite a las empresas transformar rápidamente sus procesos empresariales en aplicaciones web, distribuidas para intranet, extranet e Internet.

Con OE-BPM se aborda cada etapa del ciclo de vida para la automatización de procesos: definir, integrar, publicar, controlar, analizar, mejorar y monitorear.

OE-BPM está formado por:

Portal BPM

Es la interfaz principal para los usuarios. Allí se pueden completar tareas. Aquellos usuarios con cargos de jefatura o dueños de proceso consultan, generan reportes, y controlan procesos y recursos. Aquellos usuarios con perfil administrador modifican parámetros de configuración, administran el control de acceso de usuarios o grupos, e instalan o desinstalan procesos.

OpenEdge Business Process Modeler

OpenEdge Business Process Modeler es la herramienta del analista de procesos como parte del trabajo que realiza en la etapa de descubrimiento para lograr un modelo descriptivo del proceso [\[ver metodología -> gestión de procesos\]](#). El enfoque de trabajo es alrededor del análisis y diseño del modelo de proceso. Principalmente se crean diagramas en lenguaje estándar BPMN que representan los elementos y el flujo de control del proceso.

Progress Developer Studio for OpenEdge

Progress Developer Studio for OpenEdge es la herramienta del implementador de procesos como parte del trabajo que realiza en la etapa de implementación para lograr un modelo ejecutable del proceso [\[ver metodología -> gestión de procesos\]](#).

Aquí se implementa el código para la integración con diversas fuentes de datos para el soporte de tareas de servicio, así como la integración de la capa de presentación con el modelo ejecutable haciendo uso del API OE-BPM. Esta herramienta también la utiliza el desarrollador para la implementación de aplicaciones OpenEdge. Así, Progress Developer Studio for OpenEdge provee un entorno de desarrollo integrado basado en Eclipse en el que los usuarios pueden crear proyectos, procesos y reglas del negocio, entre otros artefactos técnicos.

BPM Webflow

Este componente permite desarrollar e implementar flujos de navegación complejos entre formularios que realizan una tarea de usuario. Hace parte del plugin BPM en Progress Developer Studio for OpenEdge.

BP Server

Es un motor de proceso. Es flexible, ligero y escalable para configuraciones tipo intranet, extranet e Internet.

BPM Events

Es un motor de reglas basado en eventos, se utiliza para formular y aplicar políticas simples en los procesos. En desarrollo se configura como parte del plugin BPM en Progress Developer Studio for OpenEdge.

Web Services

Este componente permite publicar sus procesos como servicios web. Sus configuraciones las hace un usuario administrador desde el Portal BPM.



Native Script

Es un framework de código abierto para desarrollar aplicaciones en las plataformas iOS y Android. Fué originalmente concebido y desarrollado por Progress. Las aplicaciones NativeScript se crean utilizando JavaScript o utilizando cualquier lenguaje que transpile a JavaScript, como TypeScript. NativeScript admite directamente el framework Angular y es compatible con el frame Vue a través de un complemento desarrollado por la comunidad. Las aplicaciones móviles creadas con NativeScript dan como resultado aplicaciones totalmente nativas, que usan las mismas API que si se hubieran desarrollado en Xcode o Android Studio. Además, los desarrolladores de software pueden reutilizar las bibliotecas de terceros de Cocoapods, Android Arsenal, Maven y npm.js en sus aplicaciones móviles sin la necesidad de wrappers.

3.3. Herramientas no Progress



Mongo

Es una base de datos no relacional, esto quiere decir que no hace relaciones entre tablas para extraer la información requerida por la aplicación, por ende esta base de datos es utilizada en la aplicación para almacenar grandes cantidades de datos, es el caso de la mensajería, que puede crecer exponencialmente y una base de datos no relacional permite tal tráfico. Por otra parte mongo.db es más adaptable con la mensajería push.



Socket.IO

Se puede definir como una librería que en el caso de integrity es la herramienta para manejar la mensajería push, aparte, también es útil para darle dinamismo a la aplicación, darle a integrity el manejo de información en tiempo real a través de eventos disparados en consecuencia con otro evento, que el usuario pueda ejecutar, en palabras técnicas que la aplicación se comporte como una app real time.

Cuando se trabaja con socket.io es necesario instalarlo en las dos capas de Node.js, tanto del lado del servidor como en el lado del cliente por eso se podría decir que socket.io no solo una librería si no también es un componente. Una librería instalada del lado del cliente y un componente instalado en el lado del servidor.

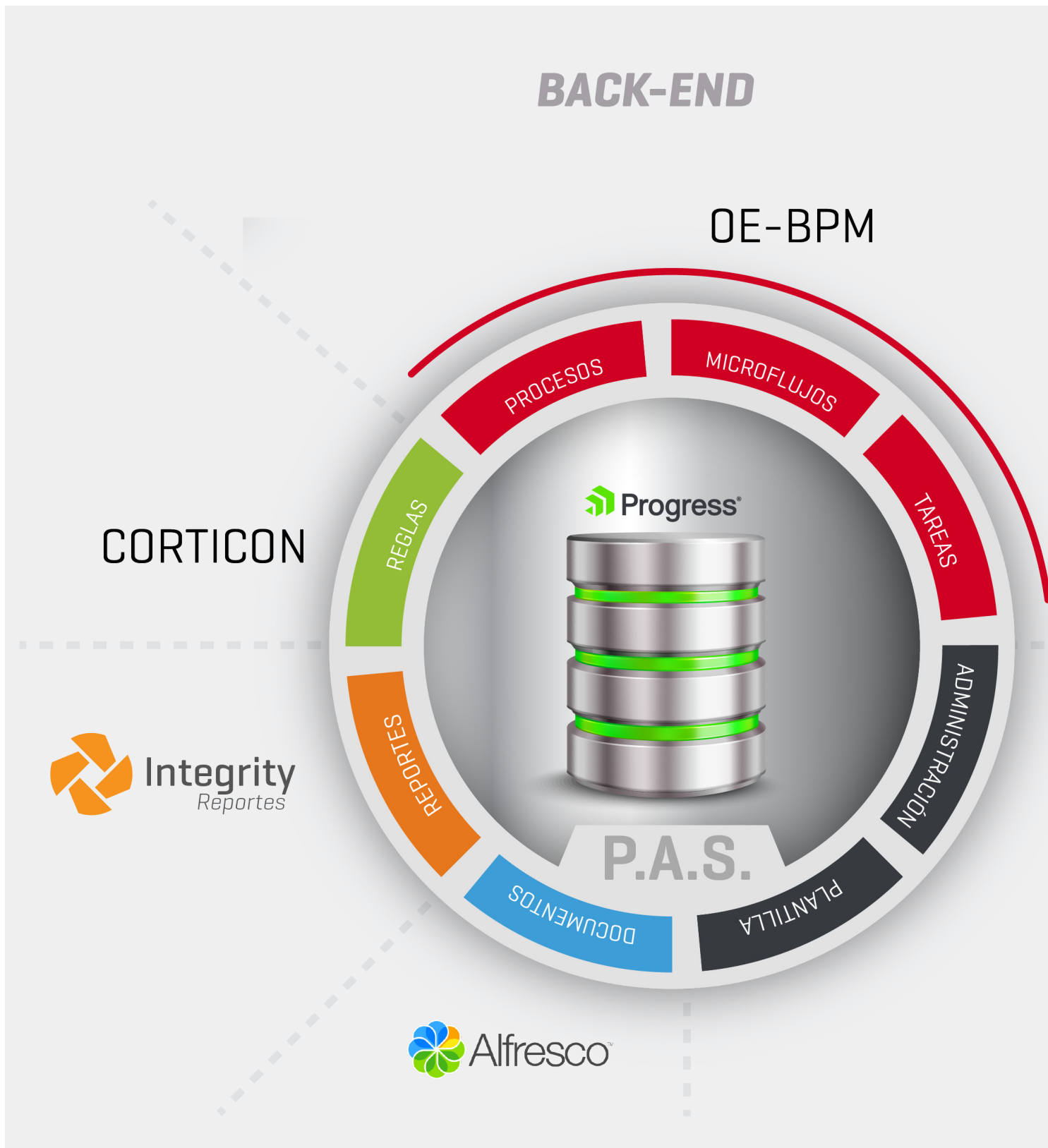
Para que los dos puedan comunicarse, es necesario hacer una conexión desde el lado del cliente y construir los diferentes eventos que se van a comunicar ya estando establecida la conexión del socket.



Alfresco

Alfresco es un sistema de administración de contenidos de código fuente libre, desarrollado en Java, basado en estándares abiertos y de escala empresarial para sistemas operativos tipo Windows, Unix Solaris y algunas versiones de Linux.

[ANEXO XIII]



MEDIA

PRESENTACIÓN

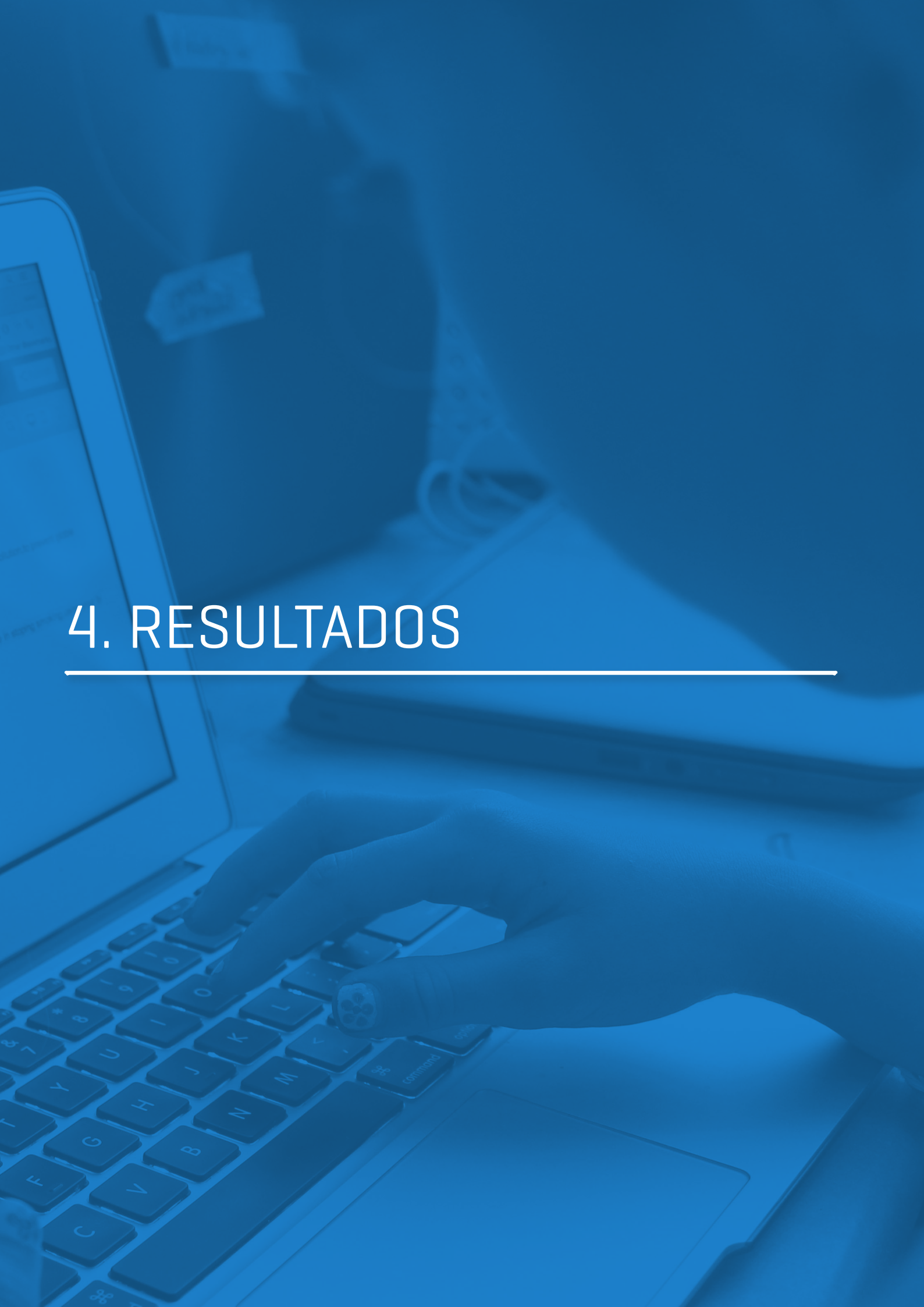


Kendo UI®



Vue.js

node
kinvey
mongoDB



4. RESULTADOS

Resultados

A continuación, este documento presentará un ejemplo de los resultados alcanzados en el proceso de modernización de "Integrity", una robusta aplicación desarrollada con el objetivo de ser la solución integral corporativa para la operatividad de la empresa.

Integrity es una aplicación originalmente desarrollada en modo caracter a la cual los usuarios pueden acceder a través de un perfil previamente definido, teniendo acceso específicamente a los procesos que cada usuario tiene permiso según su perfil y función dentro de la empresa.

Los procesos más importantes de esta aplicación se desenvuelven en el área de Nómina. La dirección y administración de los recursos comerciales y el ejercicio corporativo, administrativo y contable.

En el transcurso de los años la aplicación ha demostrado ser altamente eficiente y de suma importancia para el correcto funcionamiento de las empresas que hoy en día hacen uso de ella. Sin embargo, su interfaz de tipo caracter y su imposibilidad de acceso por internet o dispositivos móviles ha hecho que esta aplicación desarrolle una variedad de problemas y deficiencias.

El trabajo de modernización de esta robusta aplicación fue llevado a cabo por el equipo de Modernización de Aplicaciones de Quantum Data Systems, equipo que ha venido trabajando en el mejoramiento de la metodología de modernización a partir de la tecnología Progress con el fin de ofrecer a nuestros clientes las mejores oportunidades de aprovechamiento de sus sistemas informáticos e inversiones tecnológicas. Llevando el mundo de las aplicaciones empresariales al futuro.

Procedure Editor - Run

int230.p 9999.8.2.1.1.2 Formulas 19/06/2018

Formula: 01
 Descripcion: costo promedio gteneral
 Programa: inf000.p

MODIFICANDO

Nombre o det

Procedure Editor - Run

int220.p 9999.8.2.1.1.11 Transacciones 19/06/2018

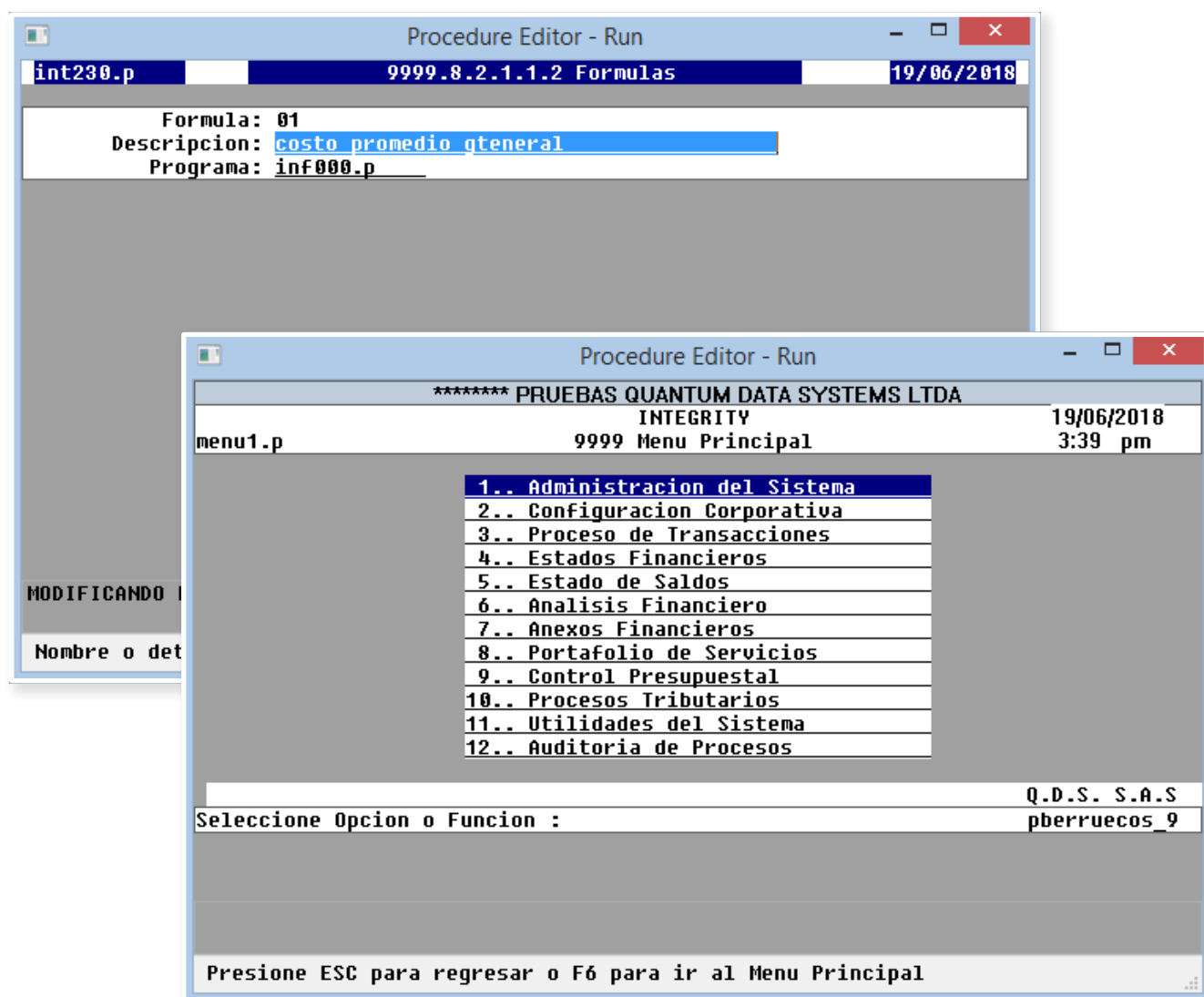
Codigo Transaccion: 005 ENTRADAS
 Concepto: 003
 Descripcion: ASIGNA PEDIDO ENTRADA
 Tipo Concepto: 1
 Clase Documento: T1 ENTREDA DE PEDIDOS
 Naturaleza (DB/CR): DB Maneja Cantidad ? : Si
 Maneja Valor ? : Si Afecta Punto Pedido ? : Si
 Requiere Labor ? : No Requiere Maquina ? : No
 Maneja Local. Destino? : No Requiere C. de Activ. : No
 Requiere Nit ? : Si Requiere Contabilizacion? : Si
 Clase Doc. de contab.: T1 ENTREDA DE PEDIDOS
 Ajuste por Inflacion? : No
 Cuenta Contable: 143501 MONTURAS
 Cuenta Contable AI:
 Requiere Ord. Compra: No Requiere Ord de Prod.? : No
 Prioridad: 1 Recosteable? : No
 Trans. a cero pesos: No Recalculable? : no
 Formato de Impresion: for100.p Columna: 1

MODIFICANDO REGISTRO

F1=Acepta F2=Ayuda ESC=Regresa F5=Borra PAGE-UP/PAGE-DOWN =

4.1. PANTALLAS ORIGINALES

La aplicación original era de tipo caracter, desarrollada en lenguaje 4GL. Su interfaz y navegación estaba muy por detrás de los estándares de desarrollo de aplicaciones.





Integrity

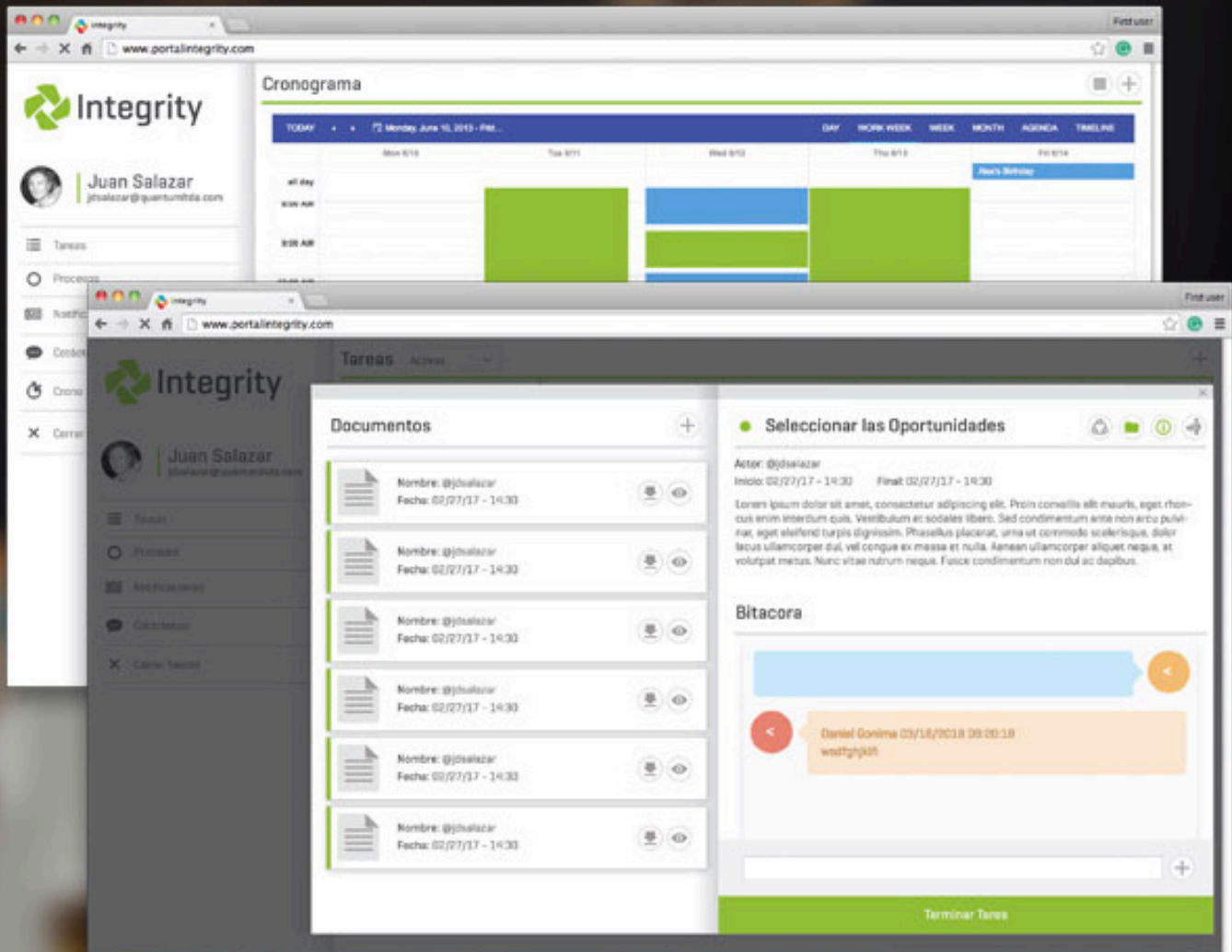
4.2. NUEVA APLICACIÓN

La nueva aplicación se aborda no solo desde el desarrollo de la interfaz, también incluye un trabajo de diseño de marca. Su nueva interfaz se adapta a dispositivos móviles.





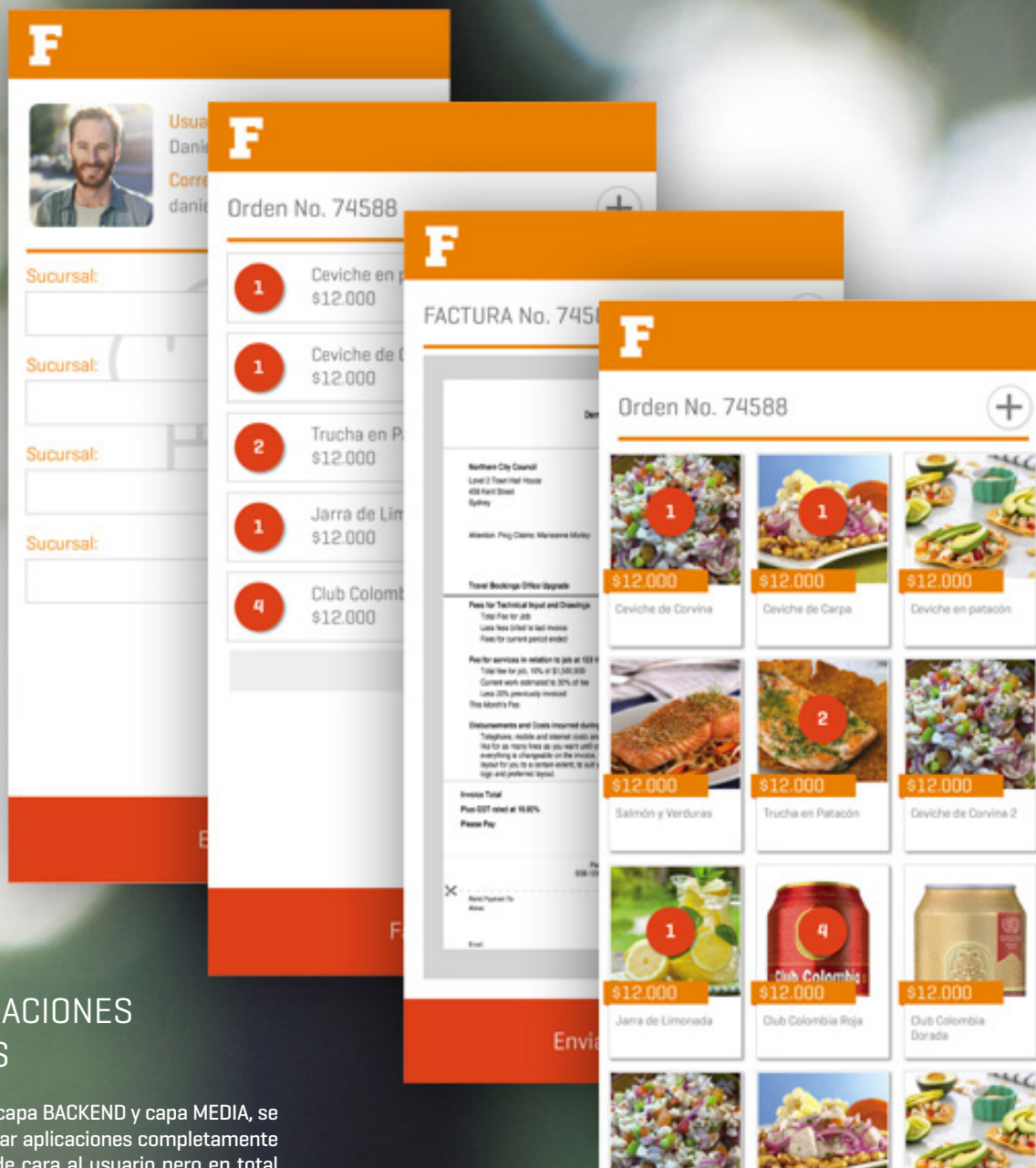
QUANTUM DATA SYSTEMS



4.3. PANTALLAS

Las nuevas pantallas ofrecen una navegación intuitiva, clara y fácil de manejar dándole al usuario un dominio completo de la herramienta con una curva de aprendizaje mucho mas positiva para los intereses de la empresa.





4.4. APLICACIONES SATÉLITES

Usando nuestra capa BACKEND y capa MEDIA, se pueden desarrollar aplicaciones completamente independientes de cara al usuario pero en total sincronía con sus masas de datos.



IntegrityLite

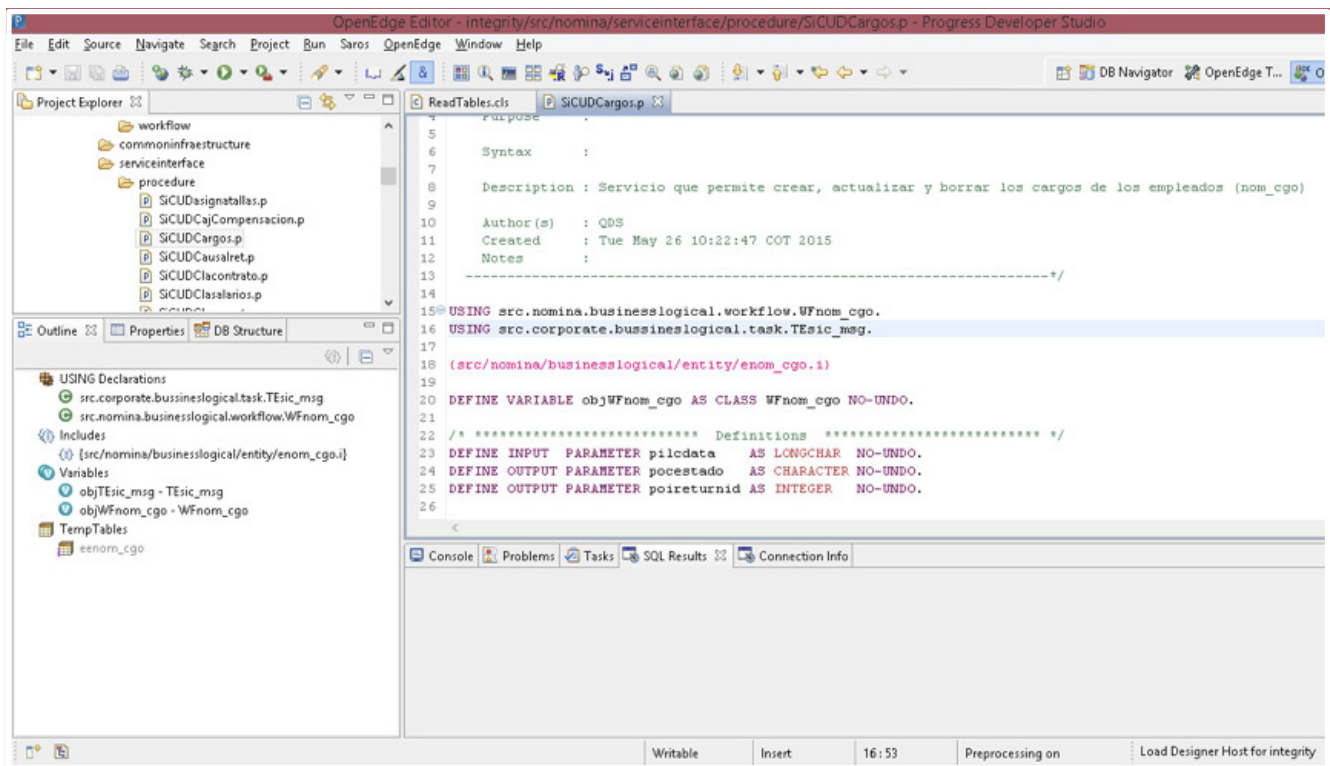
5. ANEXOS

1 2 3 4 5 6 7 8 9 10

Estado	Nombre Servicio	Definición	Parámetros																																				
			Posición Parámetro	Nombre parámetro	Tipo dato OE	Input/Output	Validaciones del parámetro	Descripción del parámetro	Ejemplo																														
ok	SiCUDJornada	Servicio que permite crear, actualizar y borrar las diferentes jornadas de nomina (nom_jor)																																					
			1	plodata	longchar	Input	En el borrado únicamente requiere el número de Jornada	<table><thead><tr><th>Nombre Elemento</th><th>Tipo de dato</th><th>Obligatorio</th><th>Descripción del elemento</th><th>Validación del elemento</th></tr></thead><tbody><tr><td>jor-rio</td><td>Integer</td><td>Si</td><td>Jornada</td><td>> 0</td></tr><tr><td>jor-des</td><td>Character</td><td>Si</td><td>Descripción</td><td>< ""</td></tr><tr><td>tmp-ola</td><td>Character</td><td>Si</td><td>Clase (Drdías, Hs)</td><td>D,H</td></tr><tr><td>jor-tmp</td><td>Decimal</td><td>Si</td><td>Tiempo de duración de la jornada</td><td>> 0</td></tr><tr><td>plindicador</td><td>Integer</td><td>si</td><td>Indicador</td><td>1Crea, 2 Actualiza, 3</td></tr></tbody></table>	Nombre Elemento	Tipo de dato	Obligatorio	Descripción del elemento	Validación del elemento	jor-rio	Integer	Si	Jornada	> 0	jor-des	Character	Si	Descripción	< ""	tmp-ola	Character	Si	Clase (Drdías, Hs)	D,H	jor-tmp	Decimal	Si	Tiempo de duración de la jornada	> 0	plindicador	Integer	si	Indicador	1Crea, 2 Actualiza, 3	<pre> { "nom_jor": "1", "nom_jor": "1", "jor-rio": 1, "jor-des": "QUINCENA", "tmp-ola": "D", "jor-tmp": 15.00, "plindicador": 0 } </pre>
			Nombre Elemento	Tipo de dato	Obligatorio	Descripción del elemento	Validación del elemento																																
			jor-rio	Integer	Si	Jornada	> 0																																
			jor-des	Character	Si	Descripción	< ""																																
			tmp-ola	Character	Si	Clase (Drdías, Hs)	D,H																																
jor-tmp	Decimal	Si	Tiempo de duración de la jornada	> 0																																			
plindicador	Integer	si	Indicador	1Crea, 2 Actualiza, 3																																			
2	pocostado	character	output	Bandera que indica si existe algun problema en el proceso																																			
3	poleturnid	int	output	>>>>>>>>>>>9, Control de Excepciones																																			
ok	SiRJornadas	Servicio que permite consultar las jornadas de la nomina; tabla nom_jor	1	plodata	longchar	Output	Archivo JSON	<table><thead><tr><th>Nombre Elemento</th><th>Tipo de dato</th><th>Obliga</th><th>Descripción del elemento</th><th>Validación del elemento</th></tr></thead><tbody><tr><td>jor-rio</td><td>Integer</td><td>Si</td><td>Jornada</td><td>> 0</td></tr><tr><td>jor-des</td><td>Character</td><td>Si</td><td>Descripción</td><td>< ""</td></tr><tr><td>tmp-ola</td><td>Character</td><td>Si</td><td>Clase (Drdías, Hs Horas)</td><td>D,H</td></tr><tr><td>jor-tmp</td><td>Decimal</td><td>Si</td><td>Tiempo de duración de la jornada</td><td>> 0</td></tr></tbody></table>	Nombre Elemento	Tipo de dato	Obliga	Descripción del elemento	Validación del elemento	jor-rio	Integer	Si	Jornada	> 0	jor-des	Character	Si	Descripción	< ""	tmp-ola	Character	Si	Clase (Drdías, Hs Horas)	D,H	jor-tmp	Decimal	Si	Tiempo de duración de la jornada	> 0	<pre> { "nom_jor": "1", "nom_jor": "1", "jor-rio": 1, "jor-des": "QUINCENA", "tmp-ola": "D", "jor-tmp": 15.00, "plindicador": 0 } </pre>					
			Nombre Elemento	Tipo de dato	Obliga	Descripción del elemento	Validación del elemento																																
			jor-rio	Integer	Si	Jornada	> 0																																
			jor-des	Character	Si	Descripción	< ""																																
			tmp-ola	Character	Si	Clase (Drdías, Hs Horas)	D,H																																
			jor-tmp	Decimal	Si	Tiempo de duración de la jornada	> 0																																
2	pljorrio	integer	Input	Parametro de entrada: 0 para ignorar el parametro, de lo contrario traera la informacion del codigo que corresponda	*																																		
4	pocostado	character	output	Bandera que indica si existe algun problema en el proceso	OK																																		
5	poleturnid	Integer	output	Lmax: 6 Lmin: 1 Formato: >>>>>>>>>>>9, Restriccion: No negativo	0																																		

Anexo I.

1. Estado del proceso si se ha desarrollado o no.
2. Nombre del proceso basado en una nomenclatura [SICUD]
3. Definición es la descripción sencilla del servicio
4. Consecutiva del parámetro en el servicio
5. Nombre del parámetro
6. Tipo del dato
7. Define si es un dato es de entrada o de salida
8. Validación del dato que entra o sale
9. Descripción sencilla del parámetro
10. Expone un ejemplo de XML o JSON; es uno de los parámetros con el que se comunica el backend con el Frontend.



Anexo II.

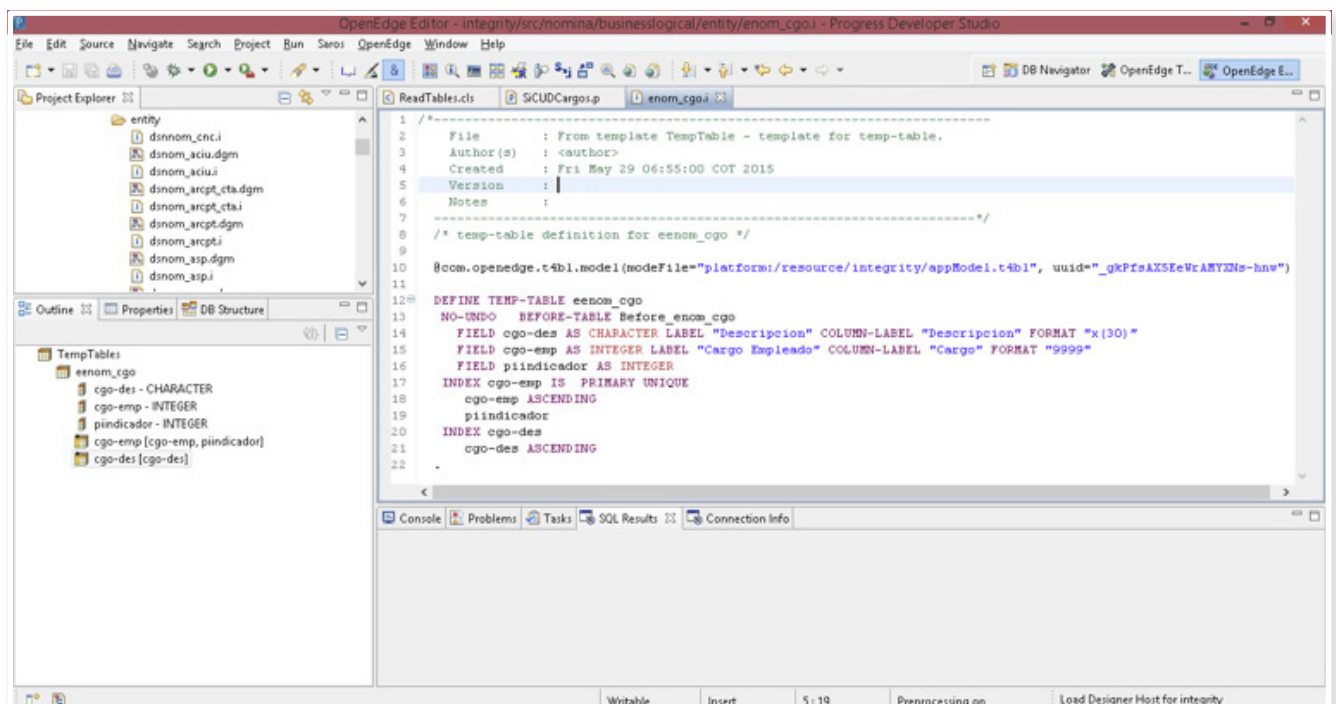
Interfaz de servicio

Para el desarrollo de esta interfaz se sugiere utilizar una variable tipo longchar. Este parámetro debe recibir un archivo json o xml con los datos definidos en la plantilla de definición de servicios y un parámetro de salida que indique el éxito o error de la ejecución del servicio.

Anexo III.

Entidades de Negocio

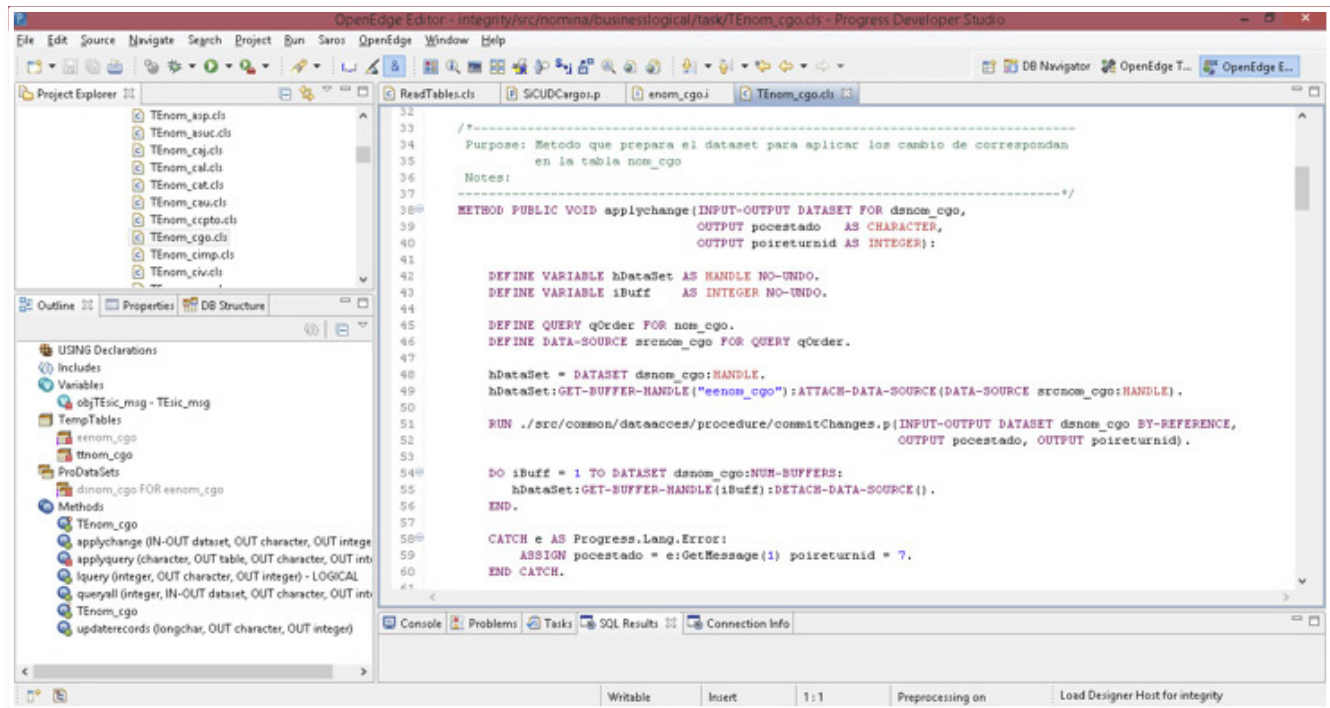
Para la entidad de negocio se sugiere definir un pro-dataset o tabla temporal, la definición de los campos no debe tener la cláusula "LIKE".



Anexo IV.

Entidad de Tarea

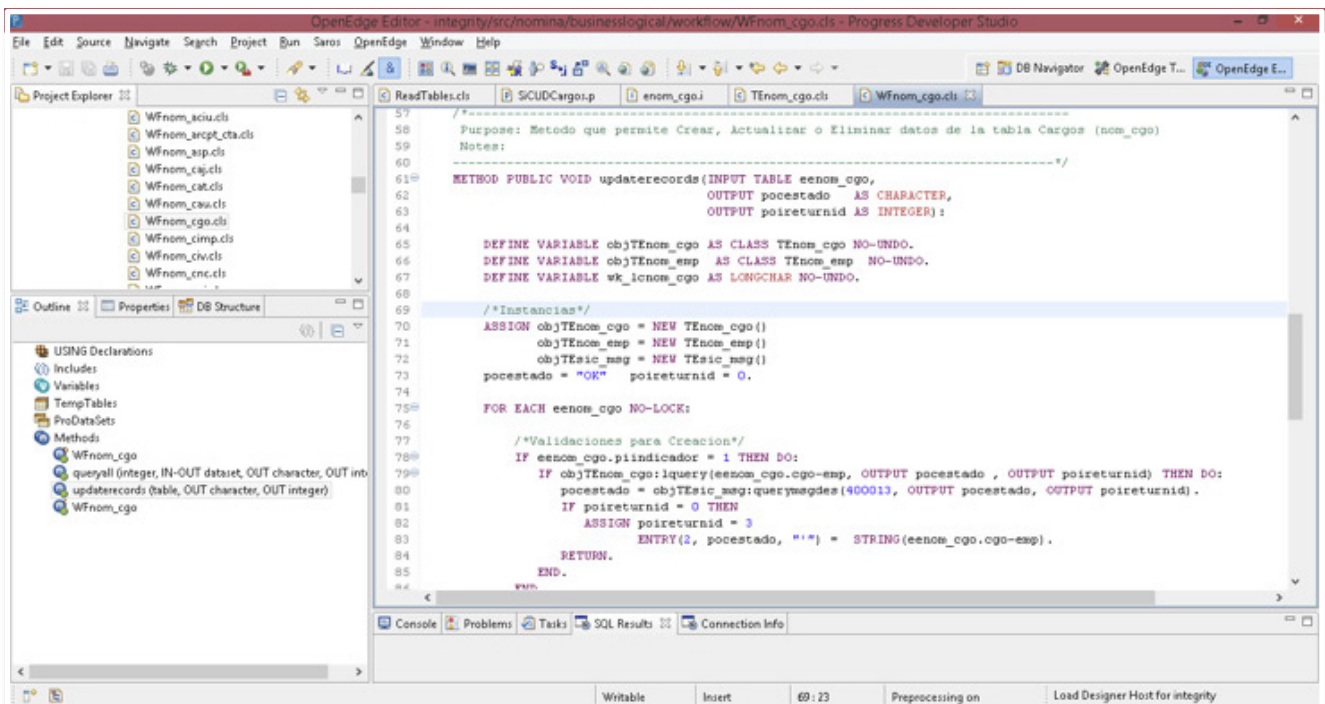
La entidad de tarea utiliza las entidades de negocio definidas, y entrega la información al flujo de trabajo mediante la tabla temporal o el pro-dataset que se haya definido.



Anexo V.

Workflow – Flujo de trabajo

El flujo de trabajo utiliza la entidad de negocio e invoca la entidad de tarea para obtener la información que requiera para cumplir con lo que se haya definido en el servicio.



Plantilla de Herramienta de Rápido Desarrollo

Desde estos campos, el usuario configura el destino de las clases y los procedimientos creados.

Desde este menú, el usuario identifica la base de datos que requiere consultar para la creación de la plantilla

QUANTUM DATA SYSTEMS

Anexo VII.

Reportes

Construya, crea, edite y comparta reportes operativos en tiempo real y sin necesidad de saber de código ni requerir de la asistencia de su departamento de tecnología.

Mis Reportes

El usuario accede a su sesión de reportes desde donde puede consultar, crear, editar o compartir.

Nuevos reportes

El usuario puede crear nuevos reportes de manera sencilla a partir de tres pasos. Creación de campos, configuración de filtros y despliegue.



Visualizaciones

Todos los datos consultados se pueden visualizar de diferentes formas, ya sean tabulares o gráficas. El usuario puede crear las gráficas que crea conveniente y construir tableros de control donde puede visualizar información de importancia en una sola pantalla y consultando los datos en tiempo real.

Anexo VIII.

Pantallas de Tareas Ad Hoc

Todas las actividades de una empresa se basan en tareas, entonces ¿Por qué no ejecutarlas de manera inteligente? Con nuestra solución los usuarios pueden crear y compartir sus tareas fácilmente.

Lista de tareas

Todo usuario tiene una lista de las tareas que tiene pendiente sin importar desde que gestor estas tareas se generan.

Opciones de tareas

Las tareas se pueden crear, compartir, hacer disponibles o finalizar a partir del menú. De igual forma, a toda tarea se le pueden adjuntar archivos y soportes.

The image displays three overlapping screenshots of the Integrity web application interface, demonstrating task management capabilities. The top-left screenshot shows the 'Cosas por hacer' (Things to do) section, listing tasks like 'Llamar a daniel para la XU tareas' and 'Hacer el mercado', each with a status (Finalizada) and a due date. The top-right screenshot shows the 'Tareas' (Tasks) section, displaying a list of tasks with details like 'Nombre: @jdsalazar' and 'Fecha: 02/27/17 - 14:30'. The bottom screenshot shows the 'Cronograma' (Calendar) view, displaying a weekly calendar with tasks scheduled as blocks, such as 'Call Charlie about the project' and 'Alex's Birthday'. Blue arrows point from the descriptive text to specific UI elements: from the 'Lista de tareas' text to the task list in the top-right screenshot, from the 'Opciones de tareas' text to the task options menu in the top-right screenshot, and from the 'Calendario' text to the calendar view in the bottom screenshot.

Anexo IX.

Micro Flujos

Es una solución que permite construir de manera sencilla procesos de negocio sin necesidad de una herramienta BPM tradicional.

Micro Flujos
El usuario construye procesos a partir de micro flujos previamente diseñado y los une como si fueran piezas de lego.

Datos del proceso

El usuario crea todos los datos que se utilizarán a lo largo del proceso. Campos de texto, numéricos, de tipo fecha entre otros.

Configuración de las tareas

El usuario puede entrar a cada tarea y configurarla al pie del detalle, definiendo los diferentes usuarios responsables por la tarea, los campos de información que requiere llenar y la información que se le debe ofrecer para poder ejecutar la tarea efectivamente.

Anexo X*.

Micro Flujos

Es una solución que permite construir de manera sencilla procesos de negocio sin necesidad de una herramienta BPM tradicional,

Condición y parámetro de comparación / Acciones a ejecutar y pruebas de valores

Vocabularios / Filtros

Corticon Designer - Vacaciones/NewRegla1.ers - Progress Developer Studio

File Edit Navigate Search Project Rulesheet Run Window Help

Proje... Rule... NewRegla1.ers

Vacaciones

DataVacaciones.xml

NewRegla1.ers

NewRegla2.ers

Regla1.ers

Regla2.ers

RFVacaciones_v1_0.eds

RFVacaciones.ers

TestNewRegla1.ert

TestNewRegla2.ert

TestRegla2.ert

TestRFVacaciones.ert

TestTRegla1.ert

Vacaciones.cdd

Vacaciones.ecore

VacVocabulary.ecore

Scope

Datatypes

dias_dinero

dias_disponibl

dias_tiempo

regla_dias_hab

Filters

1

2

3

4

5

6

7

8

Conditions

Datatypes.dias_dinero + Datatypes.dias_tiempo <= Datatypes.dias_disponibles

Actions

Post Message(s)

Datatypes.regla_dias_habiles

Overrides

Rule Operators

Attribute Operators

Entity/Association Operators

Extended Operators

General

Rule Statements

Ref ID Post Alias Text Rule Name Rule Link

1 Info Dias Solicitados dentro del Rango NewRegla1

2 Violation Dias Solicitados Fuera de Rango NewRegla1

Reglas / Atributos y operadores

Información de ejecución de la regla

QUANTUM DATA SYSTEMS

42

Reglas

Parámetros de entrada
Variables de resultados
Variables esperadas

Corticon Designer - Vacaciones/TestNewRegla1.ert - Progress Developer Studio

File Edit Navigate Search Project RuleTest Run Window Help

Quick Access Corticon Designer

Proj... Rule...

Vacaciones

- DataVacaciones.xml
- NewRegla1.ers
- NewRegla2.ers
- Regla1.ers
- Regla2.ers
- RFVacaciones_v1_0.eds
- RFVacaciones.ert
- TestNewRegla1.ert
- TestNewRegla2.ert
- TestRegla2.ert
- TestRFVacaciones.ert
- TestTRegla1.ert
- Vacaciones.cdd
- Vacaciones.ecore
- VacVocabulary.ecore

Rule Operators

- > Attribute Operators
- > Entity/Association Operators
- > Extended Operators
- > General

untitled_1

Vacaciones/NewRegla1.ers Differences: 0

Input	Output	Expected
Datatypes [1]	Datatypes [1]	
dias_dinero [10]	dias_dinero [10]	
dias_disponibles [15]	dias_disponibles [15]	
dias_tiempo [5]	dias_tiempo [5]	
	regla_dias_habiles [true]	

Rule Statements Rule Messages

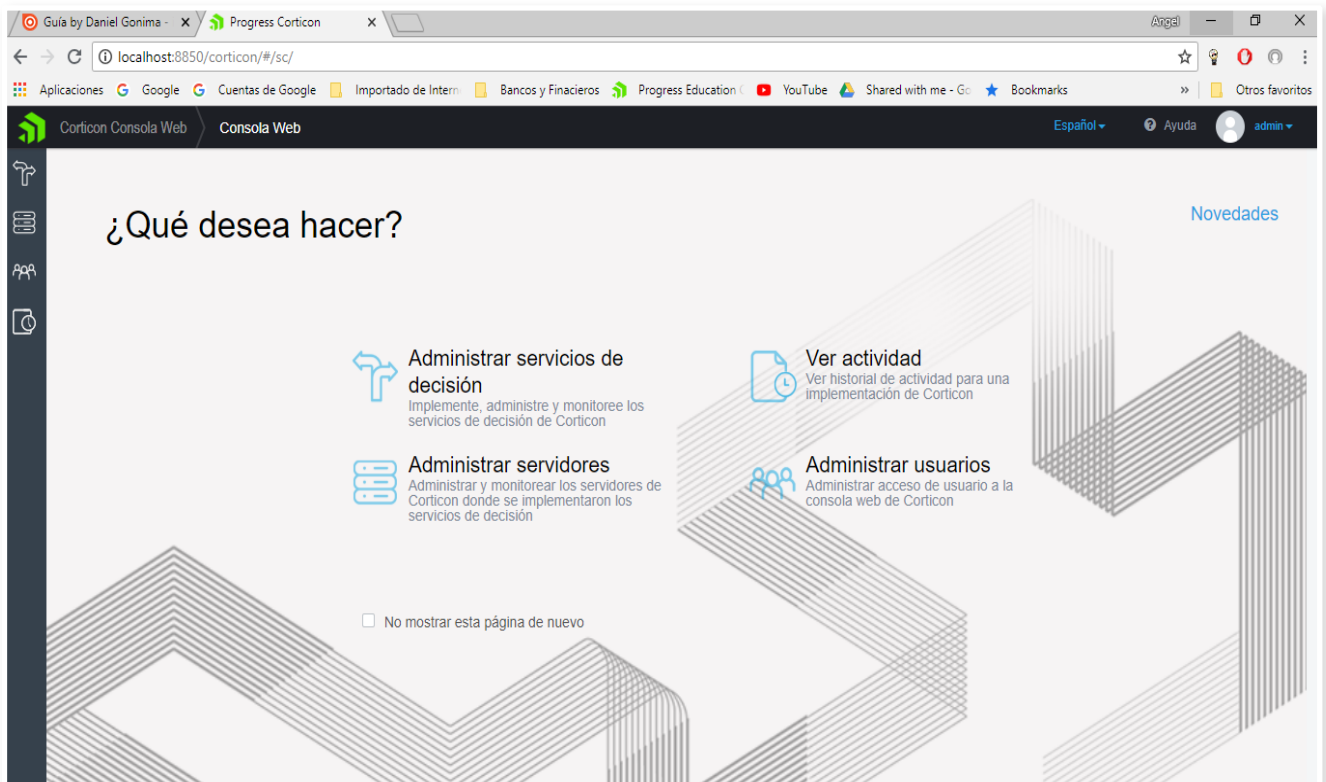
Severity	Message	Entity
Info	Dias Solicitados dentro del Rango	Datatypes[1]

Mensajes de respuesta

Anexo X*.

Micro Flujos

Es una solución que permite construir de manera sencilla procesos de negocio sin necesidad de una herramienta BPM tradicional,



Guía by Daniel Gonima - x Progress Corticon

localhost:8850/corticon/#/sc/decisionServices

Aplicaciones Google Cuentas de Google Importado de Internet Bancos y Financieros Progress Education YouTube Shared with me - Go Bookmarks Otros favoritos

Corticon Consola Web Servicios de decisión Español Ayuda admin

Servicios de decisión

+ Agregar servicio de decisión

▼ Servicios de decisión administrados

Los servicios de decisión implementados con la Consola Web se pueden administrar completamente aquí.

Nombre de servicio de decisión ▲	Versión ▼	En vigencia ▼	Expira ▼	Servidores ▼	Ejecuciones ▼	Tiempo promedio (ms) ▼
✓ DS_Vacaciones	1.0			✓ local server	0	0
✓ VacacionesIntegrity	1.0			✓ local server	0	0

1 - 2 of 2 items

Guía by Daniel Gonima - x Progress Corticon

localhost:8850/corticon/#/sc/applications/49/view

Aplicaciones Google Cuentas de Google Importado de Internet Bancos y Financieros Progress Education YouTube Shared with me - Go Bookmarks Otros favoritos

Corticon Consola Web Servicios de decisión Aplicación: DS_Vacaci... Español Ayuda admin

Detalles de aplicación: ✓ DS_Vacaciones

Editar Borrar Anular la implementación < Atrás

▼ General

Servidores:	✓ local server	Recuento de ejecuciones:	0
Implementado:	13 de ene. de 2018 13:41:37	Recuento de errores:	0
Creado por:	admin	Tiempo promedio:	0 milisegundos
Creado el:	13 de ene. de 2018 11:48:20	Última modificación:	13 de ene. de 2018 12:16:40

▼ Servicios de decisión

Nombre	Archivo Nombre	Estado	Modo de acceso a la base de datos	Recuento de ejecuciones	Recuento de errores	Tiempo promedio
DS_Vacaciones	RFVacaciones_v1_0.eds	✓	Ninguno	0	0	0 milisegundos

▼ Restricciones de los mensajes de regla

- ☐ Restringir mensajes de regla de información
- ☐ Restringir mensajes de regla de advertencia
- ☐ Restringir mensajes de regla de violación

Anexo X

La plantilla estándar es un documento en formato tabular en el cual se consigna los datos básicos del proceso: nombre, dueño de proceso, descripción, entradas y salidas. Adicionalmente, queda registro de los pasos del proceso: nombre del paso, actor o grupo responsable, descripción, instrucciones, entradas y salidas.



DEFINICIÓN DE REQUERIMIENTOS PROCESOS

NOMBRE PROCESO

DUEÑO PROCESO

DESCRIPCIÓN

ENTRADA

SALIDA

Contabilizar Soporte

Gerente Administrativo y Financiero

Contabilizar soporte permite mantener actualizados los movimientos diarios de egresos.

Soporte [cargado]

Soporte [contabilizado]

INDICE	ENTRADA	WORKSTEP	SALIDA	INSTRUCTIONS	PERFORMER	DOCUMENTATION
1	Soporte	Cargar Soporte	Soporte [cargado]	Cargue el documento legible referido a: comprobantes de egreso, soportes de inventarios, novedades de nómina. Registre todos los datos solicitados en el formulario. Registre si hubo algún abono a su factura.	Algún miembro del grupo Creator para el proceso. Nota: algún miembro del grupo Creator de Tú Óptica o del grupo Creator Integrity.	Se inicia una instancia al hacer upload de uno o más archivos que representan un soporte. Para app móvil (android) el formato del soporte debe ser JPG.
2	Soporte [cargado]	Asignar Soporte	Soporte [asignado]	Revise soporte, verifique a qué grupo pertenece y asigne. En caso de excepción, haga uso de la tarea colaborativa: donde el colaborador puede ser el creador del caso o el contador de la cuenta.	Algún miembro del grupo soporte nivel 1 Integrity.	Se lleva a cabo una validación manual del soporte cargado. En caso de timeout (duración 1h) el dueño del proceso es responsable de tomar una acción (Ej.: delegar tarea). Si : a. El soporte se puede contabilizar, entonces se selecciona el performer de acuerdo a lo especificado en la tarea Contabilizar Soporte. Si no : b. Se puede empujar una tarea colaborativa al creador de la instancia del proceso para que realice las correcciones que haya lugar. En caso de timeout (duración 5d) se puede generar un email automático al responsable de la tarea principal, quien determinará qué acción aplicar (nuevo mensaje, o, no contabilizar). c. En caso de no identificar el performer de acuerdo a lo especificado en la tarea Contabilizar Soporte, se puede empujar una tarea colaborativa a un usuario del grupo Contador Integrity. En caso de timeout (duración 1h) se puede generar un email automático al responsable de la tarea principal, quien determinará qué acción aplicar (nuevo mensaje, no contabilizar, o, notificar por algún medio al dueño del proceso).
3	Soporte [asignado]	Contabilizar Soporte	Soporte [Contabilizado]	Revise diariamente el reporte de cuadro movimiento diario y corrobore que la información coincida con los soportes cargados a la fecha. Previo a contabilizar, revise el soporte y los datos cargados en el formulario. De ser el caso, edite la información errónea cargada por la cuenta. Contabilice, registrando los datos en el formulario solicitado y presione contabilizar. Rechace en los casos de que el documento no cumpla con los datos del reporte de cuadro diario (valores no coincidan, facturas de otros proveedores, entre otros). Revise el auxiliar de cuenta, para corroborar que los saldos coincidan.	Algún miembro del grupo soporte nivel 2 Integrity, así: 1. Algún miembro del Grupo Auxiliar Contable Integrity. 2. Algún miembro del Grupo Auxiliar Inventario Integrity. 3. Algún miembro del Grupo Auxiliar Nómina Integrity.	Se lleva a cabo una validación manual (adicional) del soporte cargado. En caso de timeout (duración 1h) el dueño del proceso es responsable de tomar una acción (Ej.: delegar tarea). En caso de excepción : a. Se puede empujar una tarea colaborativa al creador de la instancia del proceso para que realice las correcciones que haya lugar. En caso de timeout (duración 5d) se puede generar un email automático al responsable de la tarea principal, quien determinará qué acción aplicar (nuevo mensaje, o, no contabilizar). b. En caso de no identificar qué procedimiento aplicar para la contabilización del soporte, se puede empujar una tarea colaborativa a un usuario del grupo Contador Integrity. En caso de timeout (duración 1h) se puede generar un email automático al responsable de la tarea principal, quien determinará qué acción aplicar (nuevo mensaje, no contabilizar, o, notificar por algún medio al dueño del proceso).

1

2

3

4

5

6

7

3. Paso:
Corresponde al nombre de un paso en el proceso, ya sea que se trate de un evento, una actividad, o un punto de decisión. Ej., cargar soporte, registrar nombre, validar información.

4. Documentación:
Se realiza una descripción en detalle del paso en el proceso.

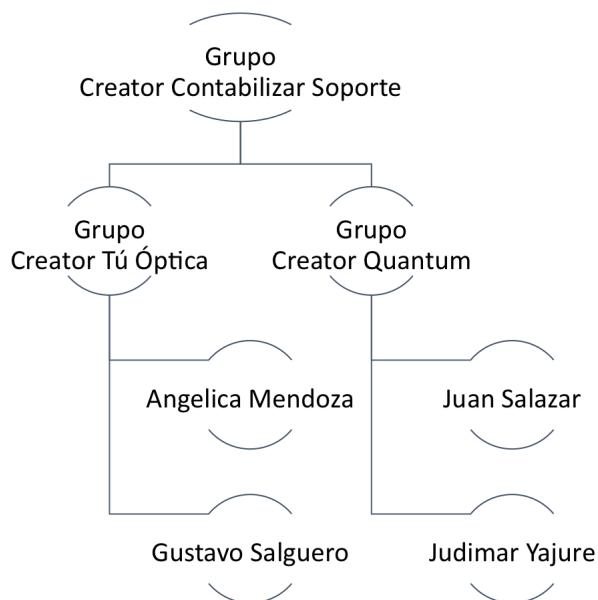
5. Instrucciones: Para actividades tipo tarea de usuario, describe las instrucciones o procedimientos que debe seguir un actor para realizar la tarea.

6. Actor: Para actividades se refiere al usuario o grupo responsable del paso en el proceso.

7. Entrada y Salida
Para actividades o eventos hace referencia a los objetos de datos o variables en el paso del proceso.

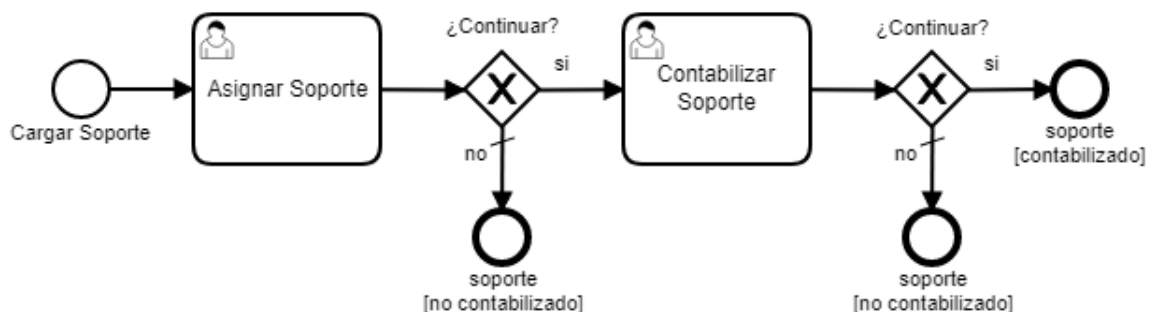
Anexo XI

Performers es un documento que captura un par de diagramas de jerarquía donde se configura los actores y grupos comprometidos con el proceso: quiénes tienen permiso para crear instancias del proceso y quiénes actúan en el rol de responsables del procesamiento de los pasos del proceso. registro de los pasos del proceso: nombre del paso, actor o grupo responsable, descripción, instrucciones, entradas y salidas.



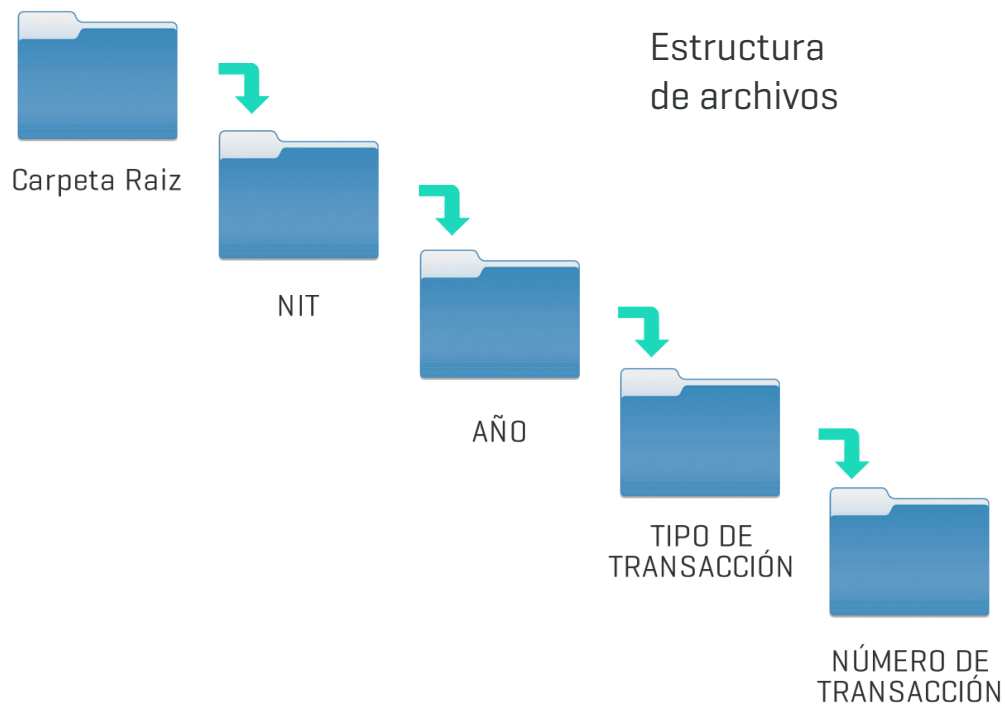
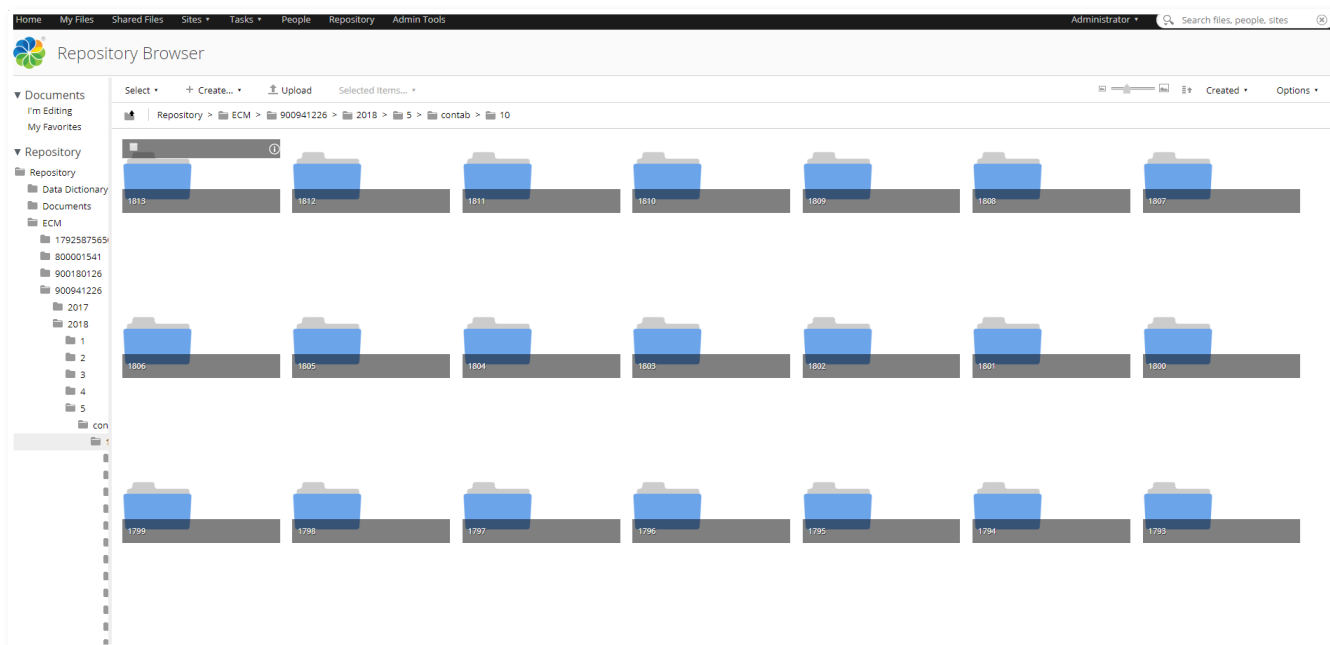
Anexo XII

En el documento 'diagrama proceso' se representa los elementos básicos del proceso de acuerdo con la notación estándar BPMN [por sus siglas en inglés: Business Process Model and Notation]. Normalmente estos elementos se clasifican como eventos, actividades, compuertas y flujos de secuencia.



Anexo XIII

Resultado final de almacén de archivos dentro de Alfresco, el usuario nunca interactúa directamente con el portal, todo se realiza automáticamente desde Integrity usando API's de Integración.





Guía básica para la
**Modernización
de aplicaciones**

Bogotá

Correo Electrónico:
quantum@quantumltda.com
Teléfonos:
PBX (57+1) 747 0518
Dirección:
Calle 97a # 60 - 50

Quito

Correo Electrónico:
quantum@quantumltda.com
Teléfonos:
PBX (593 + 2) 023237552
Dirección:
Av. Orellana E12 - 148
y 12 de Octubre