

Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Diseño de Software
Profesora Ing Ericka Solano Fernández

Proyecto 1
Documento de
Arquitectura de Software

Integrantes

Sebastián Salas Garcia 2015183511

Camila Víquez Alpízar 2014152891

Leonardo Mata Mora 2016081321

07 de Octubre del 2018

Revisión Histórica

| Fecha | Versión | Descripción | Autor |
|------------|---------|---|---|
| 06/10/2018 | 01 | Documento sobre la arquitectura del proyecto codificador y decodificador. | Camila Viquez Sebastian Salas Leonardo Mata |

Tabla de contenido

| | |
|---|-----------|
| 1. Introducción | 5 |
| Propósito | 5 |
| Alcance | 5 |
| Definiciones, acrónimos y abreviaciones | 5 |
| Referencias | 5 |
| Visión. | 6 |
| 2. Representación arquitectónica | 6 |
| 3. Objetivos y limitaciones arquitectónicas | 6 |
| 4. Vista de casos de uso | 7 |
| 4.1 Descripción de los casos de uso. | 8 |
| 4.1.1 Consultar alfabeto: | 8 |
| 4.1.2 Escribir bitácora: | 8 |
| 4.1.3 Consultar algoritmos | 9 |
| 4.1.4 Realizar codificación | 9 |
| 4.1.5 Realizar decodificación | 9 |
| 4.1.6 Consultar ayuda | 9 |
| 4.1.7 Crear algoritmos | 9 |
| 4.1.8 Crear alfabetos | 9 |
| 4.1.9 Consultar bitacoras | 9 |
| 4.1.10 Eliminar alfabeto | 9 |
| 4.1.11 Eliminar algoritmos. | 10 |
| 4.1.12 Seleccionar algoritmos. | 10 |
| 5. Vista lógica | 10 |
| 5.1 Visión. | 10 |
| 5.2 Paquetes de Diseño de gran importancia arquitectónica | 12 |
| 5.2.1 Capa de interfaz | 12 |
| 5.2.2 Capa de negocio | 12 |
| 5.2.2 Capa del Modelo | 13 |
| 5.3 Realizaciones de casos de uso | 15 |
| 6 Vista de procesos | 19 |
| 7 Vista de despliegue | 19 |
| 8 Vista de implementación | 19 |

| | |
|------------------------------------|-----------|
| 8.1 Visión de conjunto | 19 |
| 8.2 Capas | 20 |
| 9 Vista de datos (opcional) | 21 |
| 10 Tamaño y Desempeño | 21 |
| 11 Calidad | 22 |
| Extensibilidad: | 22 |
| Fiabilidad: | 22 |
| Portabilidad: | 23 |

Documento de Arquitectura de Software

1. Introducción

1.1 Propósito

Este documento proporciona una visión general de arquitectura global del sistema, usando una serie de diferentes puntos de vista de arquitectura para representar diferentes aspectos del sistema. Se pretende captar y transmitir las decisiones arquitectónicas significativas que se hayan introducido en el sistema.

1.2 Alcance

En este documento se describirán los módulos a desarrollar en el proyecto y la forma en que se interrelacionan, también serán detallados otros aspectos importantes con respecto a requerimientos no funcionales como lo son el desempeño y la calidad de la arquitectura del sistema.

1.3 Definiciones, acrónimos y abreviaciones

Socket:

Un socket es un mecanismo que permite la conexión entre distintos procesos, habitualmente se utilizan para establecer comunicaciones entre distintas máquinas que estén conectadas a través de la red.

1.4 Referencias

Juarez, M. and perfil, V. (2018). Builder. [online] Migranitodejava.blogspot.com. Available at: <http://migranitodejava.blogspot.com/2011/05/builder.html> [Accessed 8 Oct. 2018].

Codifica.me | Desarrollo web | Programación | SEO. (2018). Definición de Socket. [online] Available at: <https://www.codifica.me/definicion-de-socket/> [Accessed 8 Oct. 2018].

1.5 Visión.

Con el desarrollo de este documento se prevé influenciar, en el desarrollo del software, en el cual, se conservan la definición de requerimientos previamente establecidos y las funcionalidades, también se debe de tener en cuenta las restricciones y procedimientos para la correcta planificación del proyecto.

Se brindará una perspectiva muy amplia de la arquitectura de la aplicación, procurando que el desarrollo e interpretación de la aplicación por parte del equipo de programadores y desarrolladores sea sumamente clara y se cree el software acorde a lo establecido en el documento.

2. Representación arquitectónica

En el desarrollo del proyecto se utilizará el mecanismo de cliente servidor, en la parte del servidor se tendrá las siguientes vistas.

vista lógica

vista de proceso e implementación

vista de casos de uso

En el caso de la parte del cliente solamente se tendrá el desarrollo de las vistas de usuarios o GUI.

3. Objetivos y limitaciones arquitectónicas

Se debe de tener en cuenta algunas restricciones y requerimientos de gran importancia a la hora del desarrollo del proyecto.

- ❖ Se debe de contar con un rol de administrador en el servidor, el cual pueda realizar cambios en los alfabetos, agregar nuevos métodos de codificación y visualizar las bitácoras.
- ❖ Toda codificación o decodificación debe de registrarse en las bitácoras.
- ❖ Se debe de implementar bajo el sistema cliente-servidor, mediante el uso de sockets

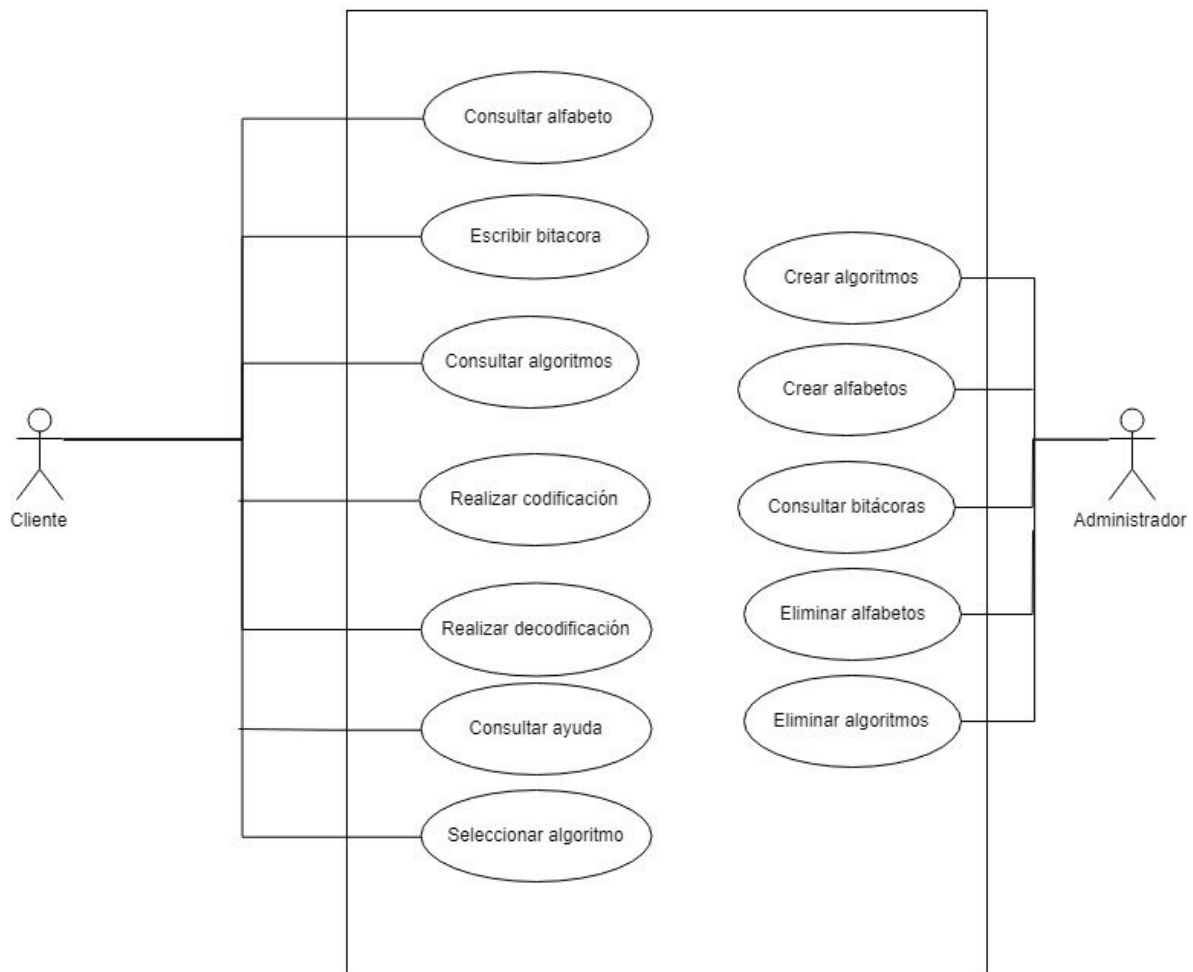
- ❖ El sistema deberá de ser desarrollado utilizando el lenguaje de programación Java.
- ❖ Los alfabetos deben de estar previamente definidos y almacenados, para su posterior uso.

4. Vista de casos de uso

En esta sección se realizará una descripción de los diferentes casos de uso del sistema, con lo cual se desarrollará una visión más amplia de los escenarios y funcionalidades del sistema.

Los casos de uso presentes en el desarrollo del codificador- decodificador son:

- Consultar alfabeto
- Escribir bitácora
- Consultar algoritmos
- Realizar codificación
- Realizar decodificación
- Consultar ayuda
- Seleccionar algoritmo
- Crear algoritmos
- Crear alfabetos
- Consultar bitacoras
- Eliminar alfabeto
- Eliminar algoritmos.



4.1 Descripción de los casos de uso.

4.1.1 Consultar alfabeto:

Breve descripción: este caso de uso permite que por parte de los usuarios clientes se realicen la consultas de los alfabetos disponibles en la aplicación.

4.1.2 Escribir bitácora:

Este caso de uso permite que los usuarios clientes seleccionen el tipo de archivo en el que desean escribir la bitácora, la cual se realizará de forma automática cada vez que el usuario realice una codificación o decodificación.

4.1.3 Consultar algoritmos

Permite que los usuarios clientes realicen la consulta de los algoritmos de codificación o decodificación disponibles para ser utilizados en la aplicación.

4.1.4 Realizar codificación

Permite que los usuarios realicen codificaciones de frases o textos, para ello deben de indicar el método a codificar, el alfabeto, la frase y en algunos casos deben de indicar la palabra clave para realizar la codificación.

4.1.5 Realizar decodificación

Permite que los usuarios realicen codificaciones de frases o textos, para ello deben de indicar el método a codificar, el alfabeto, la frase y en algunos casos deben de indicar la palabra clave para realizar la codificación.

4.1.6 Consultar ayuda

Permite que los usuarios, tengan acceso a una serie de indicaciones de como utilizar el software.

4.1.7 Crear algoritmos

Permite que el usuario administrador cree nuevos algoritmos, para la codificación y decodificación de texto.

4.1.8 Crear alfabetos

Permite que el usuario administrador cree nuevos alfabetos de símbolos permitidos en el ingreso de las frases para la codificación o decodificación

4.1.9 Consultar bitacoras

Este caso de uso permite que el usuario administrador consulte las bitácoras creadas en las ejecuciones del software.

4.1.10 Eliminar alfabeto

Permite que el usuario administrador elimine alfabetos para codificación y decodificación los cuales ya se encuentran creados en el software.

4.1.11 Eliminar algoritmos.

Permite que el usuario administrador elimine algoritmos para codificación y decodificación de textos, los cuales ya se encuentra creados en el software.

4.1.12 Seleccionar algoritmos.

Permite que los usuarios seleccionen uno o varios algoritmos de los algoritmos registrados previamente por el administrador.

5. Vista lógica

5.1 Visión.

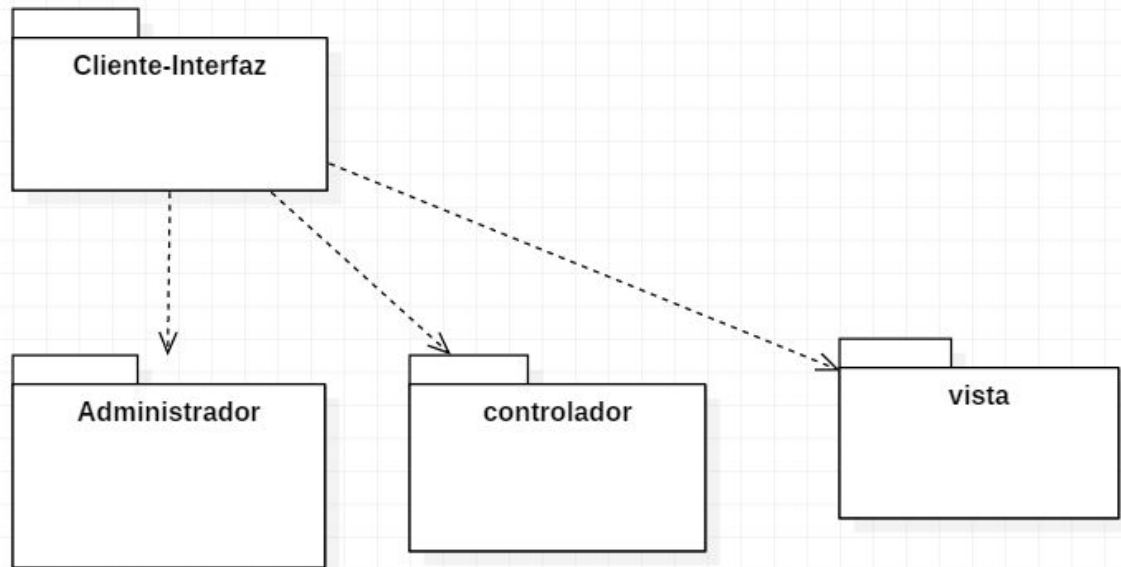
La siguiente sección describe el modelo y organización lógica del sistema desarrollado. Se incluyen los diagramas de paquetes y clases principales clasificados en capas según su función en el software.

Adicionalmente, se incluyen las realizaciones de algunos de los casos de uso principales para demostrar las dinámicas importantes del sistema.

El sistema Codificador Decodificador está constituido en dos módulos ya que uno representa el cliente y otro el servidor.

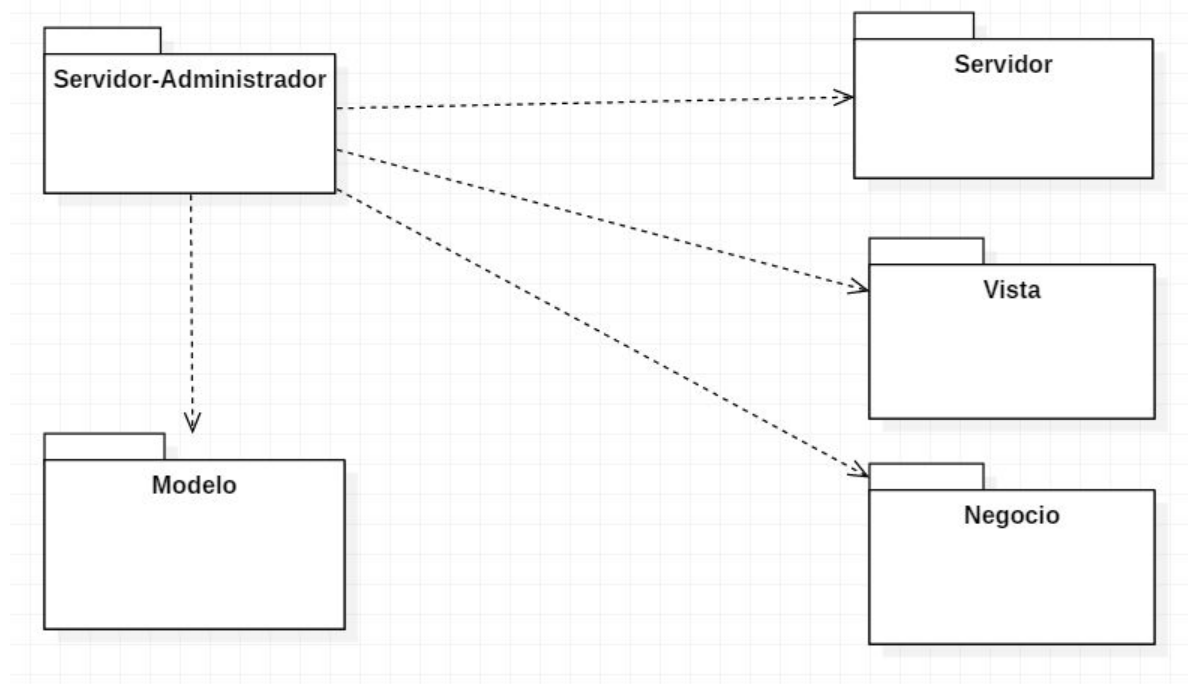
Dentro de las capas o paquetes principales del cliente se encuentran las siguientes:

- **Administrador:** Contiene la clase cliente, que es la encargada de almacenar los atributos necesarios para establecer la conexión por medio de sockets con el servidor. También contiene la clase AdmCliente quien se encarga de la comunicación del DTO entre el servidor y el cliente.
- **Controlador:** Dentro de este paquete se encuentra la clase encargada de crear los objetos a partir de la información recibida por el usuario por medio de la interfaz, también se encarga de mostrar los datos en la interfaz para que el usuario los pueda ver.
- **Vista:** Contiene la clase que contiene el main, para ejecutar el cliente. También contiene la interfaz gráfica.



El otro módulo, mencionado anteriormente, comprende el servidor este está dividido en cuatro paquetes o capas principales.

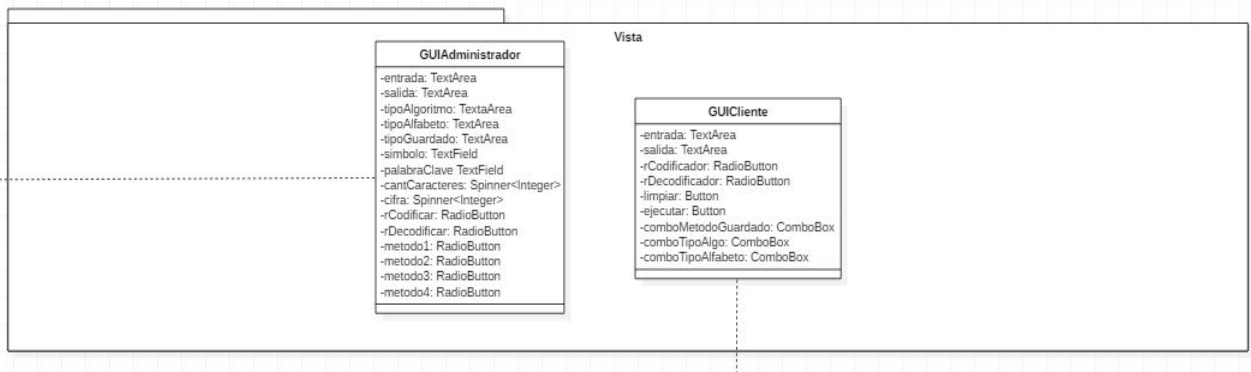
- **Modelo:** Contiene las clase necesarias para los algoritmos de codificación y decodificación.
- **Negocios:** Dentro de esta capa se encuentra la clase controlador, encargada de la llamada a métodos y la activación de los algoritmos de codificación y decodificación. Cuenta con la clase DAOTXT que es la encargada de persistir los datos correspondientes a los alfabetos. También maneja las clase que cumple con la funcionalidad de gestores tanto de alfabetos como de información necesaria dentro de la aplicación.
- **Vista:** Esta capa cuenta con la implementación de la clase administradora del cliente, esta es la clase encargada del manejo de datos e interfaces que son mostradas al usuario.
- **Servidor:** Esta capa es la encargada de crear los sockets y la comunicación entre el cliente y la aplicación. En ella se especifican los puertos necesarios para dicha comunicación.



5.2 Paquetes de Diseño de gran importancia arquitectónica

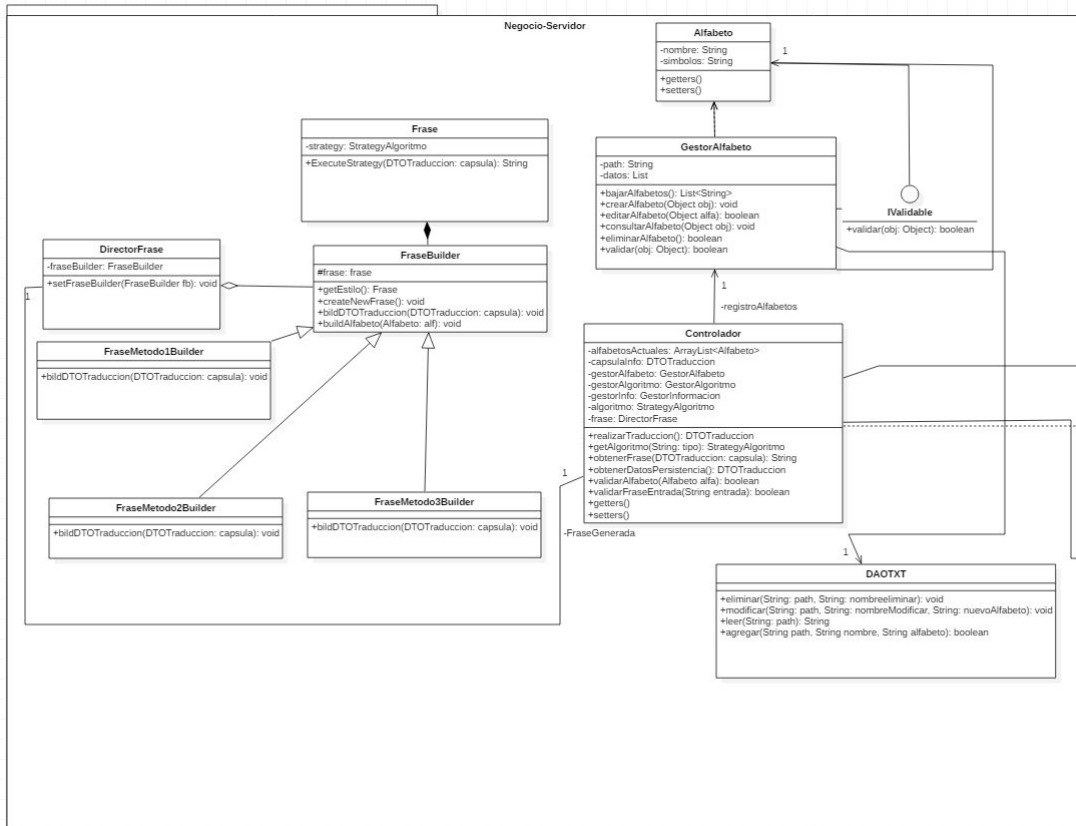
5.2.1 Capa de interfaz

A continuación se presentan unas capturas de pantalla referentes a las clases pertenecientes a la capa de interfaz.



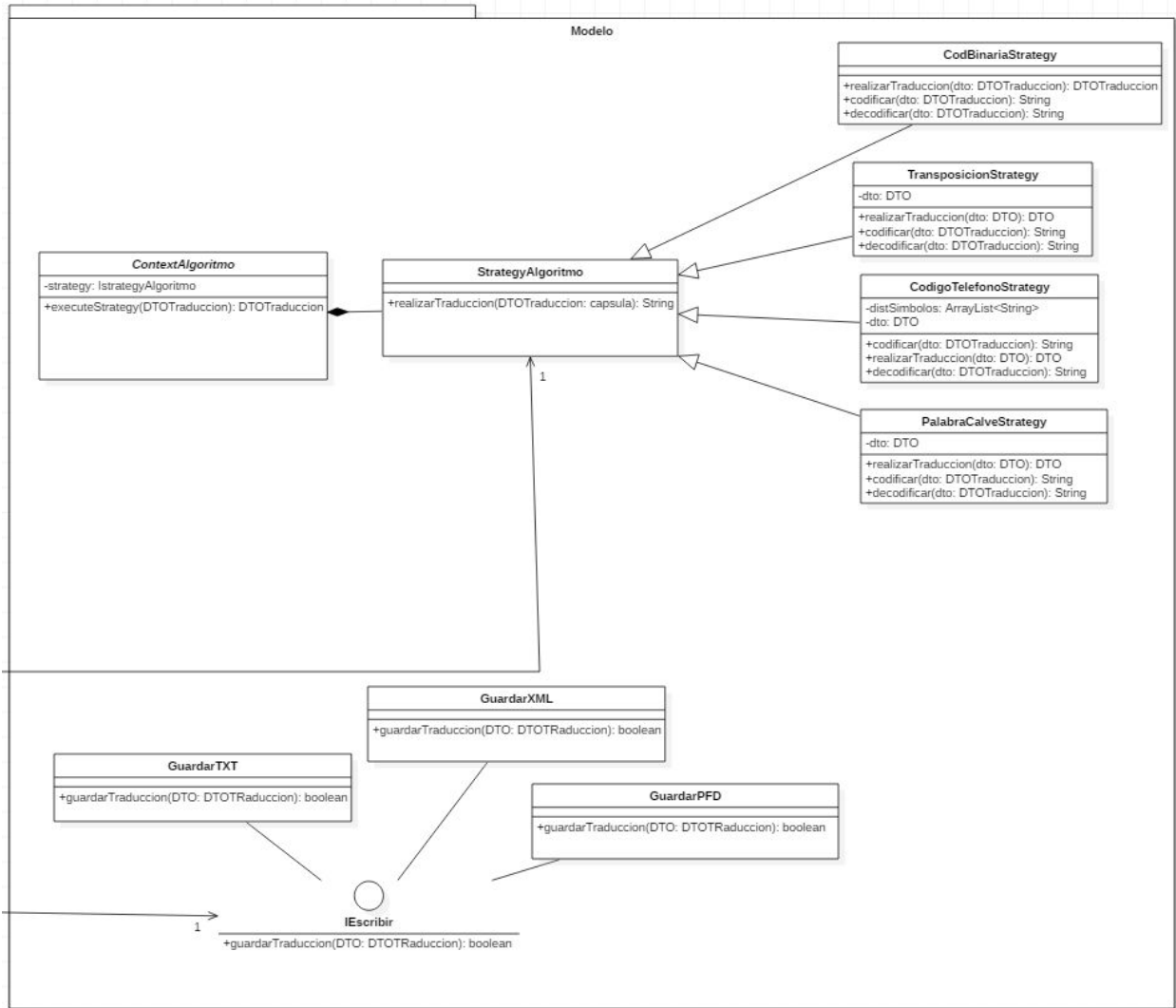
5.2.2 Capa de negocio

A continuación se presenta una secuencia de imágenes que ejemplifican las clases implementadas en la capa del negocio.



5.2.2 Capa del Modelo

A continuación se presenta una secuencia de imágenes que ejemplifican las clases implementadas en la capa del modelo.



5.3 Realizaciones de casos de uso

| | | |
|-------------------------|---|---|
| RF- 02 | Realizar codificación | |
| Versión | 01 | |
| Autores | Usuario regular | |
| Descripción | El sistema deberá de registrar en una bitácora las codificaciones que realice el usuario. | |
| Precondición | Selección del alfabeto y el algoritmo de codificación. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario regular debe de seleccionar el tipo de acción a realizar “codificar, decodificar” |
| | 2 | El usuario regular debe de seleccionar el tipo de archivo que desea para la bitácora, “txt, PDF, XML” |
| | 3 | El usuario regular debe de seleccionar los algoritmos de codificación que desee aplicar sobre la frase. |
| | 4 | El usuario regular debe de seleccionar el alfabeto |
| | 5 | El usuario regular presiona el botón ejecutar |
| | 6 | El sistema envía los datos por medio del DTO al servidor para la consulta respectiva. |
| | 7 | El sistema recibe los datos y realiza los algoritmos de codificación. |
| | 8 | El sistema realiza la escritura de la frase sin codificar y la frase codificada en el archivo del tipo seleccionado por el usuario. |
| | 9 | El sistema por medio de su servidor devuelve al usuario regular el DTO con la frase codificada |
| Postcondición | Se mostrará en la vista del cliente la frase codificada. | |
| Excepciones | Paso | Acción |
| | 1 | Si el usuario no selecciona el tipo de acción “codificar o decodificar”, se le mostrará un mensaje de error. |
| | 2 | Si el usuario regular no selecciona un alfabeto, se mostrará una ventana de error. |
| | 3 | Si el usuario regular no ingresa caracteres válidos dentro del alfabeto, se mostrará una ventana de error. |
| | | |
| Frecuencia | 5 veces / minuto | |

| | |
|--------------------|----------------|
| esperada | |
| Importancia | Vital |
| Urgencia | Inmediatamente |
| Comentarios | |

| | | |
|-------------------------|---|---|
| RF- 07 | Realizar decodificación | |
| Versión | 01 | |
| Autores | Usuario regular | |
| Descripción | El sistema deberá de registrar en una bitácora las decodificaciones que realice el usuario. | |
| Precondición | Selección del alfabeto y el algoritmo de codificación. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario regular debe de seleccionar el tipo de acción a realizar “codificar, decodificar” |
| | 2 | El usuario regular debe de seleccionar el tipo de archivo que desea para la bitácora, “txt, PDF, XML” |
| | 3 | El usuario regular debe de seleccionar los algoritmos de codificación que desee aplicar sobre la frase. |
| | 4 | El usuario regular debe de seleccionar el alfabeto |
| | 5 | El usuario regular presiona el botón ejecutar |
| | 6 | El sistema envía los datos por medio del DTO al servidor para la consulta respectiva. |
| | 7 | El sistema recibe los datos y realiza los algoritmos de decodificación. |
| | 8 | El sistema realiza la escritura de la frase sin codificar y la frase codificada en el archivo del tipo seleccionado por el usuario. |
| | 9 | El sistema por medio de su servidor devuelve al usuario regular el DTO con la frase codificada |
| Postcondición | Se mostrará en la vista del cliente la frase codificada. | |
| | Paso | Acción |
| | 1 | Si el usuario no selecciona el tipo de acción “codificar o decodificar”, se le mostrará un mensaje de error. |
| | 2 | Si el usuario regular no selecciona un alfabeto, se mostrará una ventana de error. |

| | | |
|----------------------------|------------------|--|
| Excepciones | 3 | Si el usuario regular no ingresa caracteres válidos dentro del alfabeto, se mostrará una ventana de error. |
| | | |
| Frecuencia esperada | 5 veces / minuto | |
| Importancia | Vital | |
| Urgencia | Inmediatamente | |
| Comentarios | | |

| | | |
|----------------------------|---|--|
| RF- 06 | Consultar ayuda | |
| Versión | 01 | |
| Autores | Usuario regular | |
| Descripción | El sistema le brindará información de cómo realizar la codificación o decodificación. | |
| Precondición | | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario regular debe de presionar el botón de ayuda. |
| | 2 | Se les desplegará una ventana a cual contiene la información de ayuda para los usuarios. |
| Postcondición | El usuario deberá de cerrar la ventana, para seguir usando el software | |
| Excepciones | Paso | Acción |
| | | |
| Frecuencia esperada | Poca | |
| Importancia | Vital | |
| Urgencia | Inmediatamente | |
| Comentarios | | |

| | |
|---------------|------------------------------|
| RF- 11 | Seleccionar algoritmo |
|---------------|------------------------------|

| | | |
|----------------------------|--|--|
| Versión | 01 | |
| Autores | Usuario regular | |
| Descripción | El sistema deberá de registrar en una bitácora las codificaciones ó decodificaciones que realice el usuario. | |
| Precondición | El administrador haya definido los algoritmos en el programa | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario regular debe de seleccionar el tipo de acción a realizar “codificar, decodificar” |
| | 2 | El usuario regular debe de seleccionar el tipo de archivo que desea para la bitácora, “txt, PDF, XML” |
| | 3 | El usuario regular debe de seleccionar los algoritmos de codificación que desee aplicar sobre la frase, ya sea uno o varios. |
| Postcondición | Deberá de ingresar la frase a codificar y ejecutar el programa. | |
| Excepciones | Paso | Acción |
| | 1 | Si el usuario no selecciona el tipo de acción “codificar o decodificar”, se le mostrará un mensaje de error. |
| | 2 | Si el usuario regular selecciona un alfabeto no válido se le mostrará una ventana de error. |
| Frecuencia esperada | Alta | |
| Importancia | Vital | |
| Urgencia | Inmediatamente | |
| Comentarios | | |

6 Vista de procesos

La aplicación no cuenta con la implementación de hilos, sin embargo, la aplicación funciona bajo el sistema cliente servidor con el uso de sockets.

Las computadoras se conectan entre sí y permiten el envío de objetos tipo DTO, en la vista del cliente se envían el objeto al servidor, una vez que este es recibido, el servidor se encarga de realizar el análisis correspondiente para determinar la acción a realizar ya sea codificar o decodificar, realiza la verificación de que los datos ingresados por usuarios son correctos y en caso de encontrar un error se lo notifica al usuario, por ejemplo si el usuario ingresa un carácter inválido, el sistema le indicará al usuario que los caracteres no son válidos, si todas las validaciones son correctas se procede a realizar la codificación o decodificación correspondiente, luego de realizarla se devuelve a la vista del cliente un usuario de tipo DTO, en el cual le envía la frase codificada, una vez que se recibe en la vista del cliente se muestra la frase procesada en la interfaz.

7 Vista de despliegue

A continuación se presenta una lista de los requerimientos de hardware necesarios para ejecutar el software:

- Sistema operativo Windows 7 o versiones superiores.
- NetBeans IDE 8.2
- Java SE 8u181
- Librería itextpdf-5.4.1.jar

8 Vista de implementación

Esta sección describe la estructura general del modelo de implementación, la descomposición de la aplicación en capas y subsistemas en el modelo de implementación y cualquier componente arquitectónicamente significativo.

8.1 Visión de conjunto

El proyecto Codificador Decodificador está compuesto por dos módulos, uno representa el cliente que contiene tres paquetes principales y otro corresponde al servidor que tiene cuatro paquetes. Puede consultar el diagrama de capas en la sección 5 de este documento.

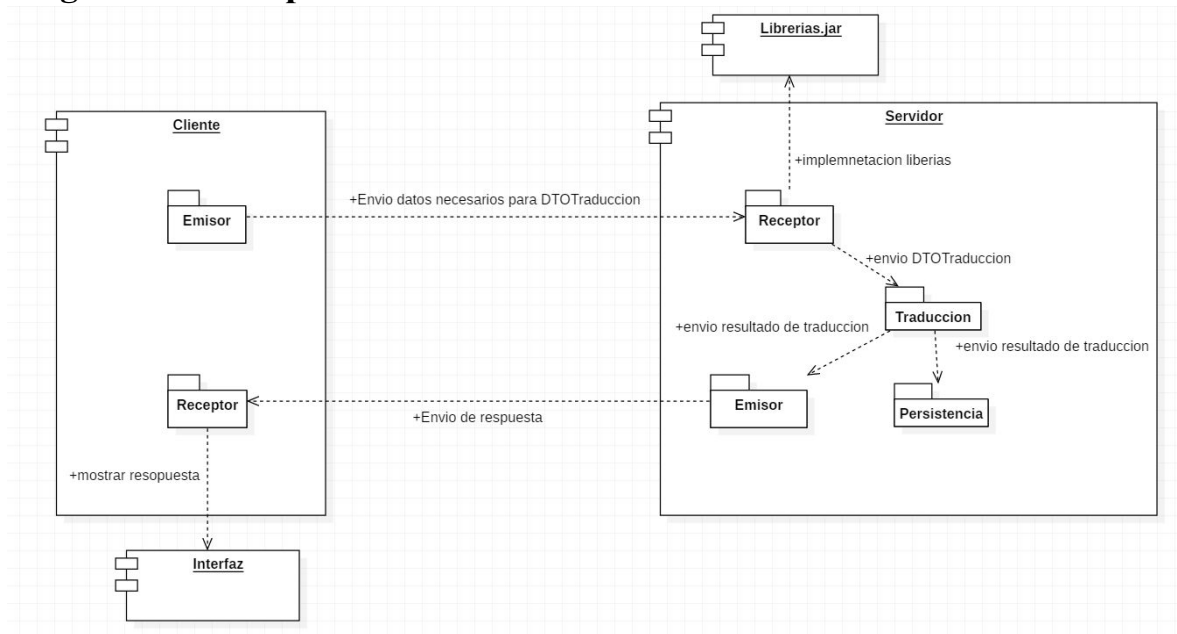
8.2 Capas

En esta sección se presenta cada capa y los subsistemas con los que interactúa.

| Proyecto Cliente | | |
|------------------|--|------------------|
| Capa | Subsistema con el que interactúa | Componente |
| Administrador | <ol style="list-style-type: none"> Administrador cliente Cliente | Cliente-Interfaz |
| Controlador | <ol style="list-style-type: none"> GuiClienteControlador | |
| Vista | <ol style="list-style-type: none"> Ejecutar Cliente | |

| Proyecto Servidor | | |
|-------------------|--|------------|
| Capa | Subsistema con el que interactúa | Componente |
| Modelo | <ol style="list-style-type: none"> Alfabeto CodigoBinario CodigoPalabraClave CodigoTelefónico CódigoTransposición CodigoVigenere ContextAlgoritmo GuardarPDF GuardarTXT GuardarXML IEscribir StrtaegyAlgoritmo | Servidor |
| Negocios | <ol style="list-style-type: none"> Controlador DAOTXT GestorAlfabeto GestorFrases GestorInformacion IValidable | |
| Vista | <ol style="list-style-type: none"> FRServidor GUIAdmin | |
| Servidor | <ol style="list-style-type: none"> Servidor | |

Diagrama de componentes



9 Vista de datos (opcional)

En esta sección se detallan los medio implementados para la persistencia de datos en la aplicación. Dentro de las funcionalidades de la aplicación es necesario almacenar los alfabetos esto se lleva a cabo mediante la implementación de archivos de texto.

La aplicación permite que el usuario seleccione entre archivos XML, PDF o TXT para guardar las traducciones realizadas durante la ejecución del programa.

10 Tamaño y Desempeño

A continuación se enumeran las características y limitaciones de rendimiento y tamaño del sistema:

- a. Conexión por medio de sockets:
 - i. El número máximo de conexiones simultáneas corresponden a un cliente. El hecho de manejar un único cliente implica un menor uso de recursos por lo que las implicaciones a nivel de memoria no son altas y el sistema no va a presentar problemas de rendimiento por este factor.

b. Persistencia de alfabetos:

- i. Los alfabetos se almacenan en archivos .txt .esto requiere de 112 bytes o más por lo que es muy conveniente el uso de este tipo de archivos para el almacenamiento de datos.

11 Calidad

A continuación se presenta una descripción de las características de calidad que brinda el sistema:

Extensibilidad:

El sistema presenta la implementación de patrones, de desarrollo de software, que hacen posible la extensibilidad del código. Se ha implementado un patrón Builder, para el manejo de las frases de entrada ya que el usuario puede determinar ciertas características para crearlas. Si el programa tuviera la necesidad de ingresar otras características estas solo deben ser agregadas como clases hijas en el patrón Frase Builder y este seguirá funcionando sin ninguna otra modificación.

Por otro lado se ha implementado el patrón strategy, implementado para los algoritmos encargados de codificar y decodificar. Esto es altamente extensible ya que el patrón recibe el nombre del método y crea una nueva instancia del método deseado.

Un punto importante dentro de este programa que favorece a la extensibilidad es el uso de la introspección para el manejo de los alfabetos y los algoritmos de codificación y decodificación, esto hizo posible simplificar la lógica del código en un alto nivel y permitir una mayor extensibilidad para futuros cambios.

Fiabilidad:

Las probabilidades de que el sistema realice las funcionalidades requeridas por el usuario solo dependen de la conexión por medio de sockets, este es el factor más

vulnerable que tiene el programa. Se puede decir que el programa es altamente fiable ya que lo mencionado anteriormente no es altamente probable.

Portabilidad:

El programa es portable ya que se implementa en java y el bytecode generado puede ser ejecutado en cualquier hardware siempre y cuando se tenga JVM compatible.