



UNIVERSIDAD JOSÉ ANTONIO PÁEZ

**DESARROLLO DE SOFTWARE PARA
TRAZADOS DE RADIOGRAFÍAS
ORTODÓNTICAS PARA LA FACULTAD
DE ODONTOLOGÍA DE LA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FASE II**

Autor: Luis Alejandro González Rico

C.I. V-24.423.007

Urb. Yuma II, Calle N°3, Municipio San Diego
Teléfono: (0241) 8714240 Master- Fax (0241) 8712339



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN**

**DESARROLLO DE SOFTWARE PARA TRAZADOS DE RADIOGRAFÍAS
ORTODÓNTICAS PARA LA FACULTAD DE ODONTOLOGÍA DE LA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ FASE II**

Trabajo de grado presentado como requisito para optar al título de

INGENIERO EN COMPUTACIÓN

Autor: González Rico Luis Alejandro

C.I. V- 24.423.007

Tutor Académico: MSc. Oneida Jiménez

C.I. V-10.227.464

San Diego, Junio de 2017



Universidad José Antonio Páez
Facultad de Ingeniería

FI-TG-2017-1CR-050

Valencia, 13 de Enero de 2017.

Ciudadano
González Luis
C.I. 24.423.007
Presente.-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 1-2017 de fecha 13/01/2017 aprobó el proyecto de trabajo de grado titulado "**DESARROLLO DE SOFTWARE PARA TRAZADOS DE RADIOGRAFIAS ORTODONTICAS PARA LA FACULTAD DE ODONTOLOGIA DE LA UNIVERSIDAD JOSE ANTONIO PAEZ. FASE II**" Presentado por usted como requisito para optar al título de Ingeniero en Computación.

Se ratifica la designación de la Ing. Oneida Jiménez, C.I. 10.227.464 y la Ing. Alicia Pizzella, C.I. 4.598.880 como Tutotes Académicos que lo asesorarán en el desarrollo de este proyecto.

Atentamente,


Prof. Marlene Zambrano
Decana (Encargada) de la Facultad de Ingeniería
(C/502 de fecha 11/10/2016)



c. c. Coordinación de Pasantías y Trabajo de Grado (2).
Archivo.

MEZ/jp



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN**

ACEPTACIÓN DEL TUTOR

Quien suscribe, Magister Oneida Jiménez portador de la cédula de identidad N° 10.227.464, en mi carácter de tutor del trabajo de grado presentado por el ciudadano Luis Alejandro González Rico, portador de la cédula de identidad N° 24.423.007, titulado **DESARROLLO DE SOFTWARE PARA TRAZADOS DE RADIOGRAFÍAS ORTODÓNTICAS PARA LA FACULTAD DE ODONTOLOGÍA DE LA UNIVERSIDAD JOSÉ ANTONIO PÁEZ FASE II**. Presentado como requisito parcial para optar al título de Ingeniero en Computación, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los 08 días del mes de junio del año dos mil diecisiete.


MSc. Oneida Jiménez
C.I.: V-10.227.464



DEDICATORIA

Primeramente, le agradezco a Dios por concederme la oportunidad de realizar este trabajo de grado para optar por mi título de Ingeniero en Computación, gracias a Él por haberme guiado y apoyado en cada situación durante este largo trayecto.

Seguidamente les agradezco a mis padres Zuleima Rico y Luis González por haberme acompañado durante toda mi carrera, a pesar de todos los problemas que se pudieron haber presentado, ellos fueron un pilar fundamental para llegar al final de la misma.

A mis compañeros y amigos que hice durante la carrera que nos apoyamos mutuamente a lo largo de los años, que a pesar de los disgustos que se podían generar juntos logramos alcanzar la meta que nos trazamos.

A mis profesores que sirvieron de guía para llegar al final de la mejor manera posible, aquellos profesores que más allá de dar una clase se preocupaban por que aprendiéramos preparándonos para un futuro.

Luis Alejandro González Rico

AGRADECIMIENTOS

Mi principal agradecimiento se lo debo a Dios, por siempre estar a mi lado, acompañándome y guiándome esperando siempre lo mejor de mí.

A mi tutora MSc. Oneida Jiménez por apoyarme a lo largo del estudio de ésta carrera, por enseñarme lo mejor para poder desenvolverme como un profesional integro, por exigirme tanto como profesional como de manera personal, gracias por brindarme las herramientas adecuadas para llegar a este punto en las mejores condiciones. Gracias por todo el apoyo incondicional que brinda tanto dentro como fuera del campus universitario.

A los profesores Ing. Mayerlin Maldonado, Ing. Luis Aguilar, Ing. Juan Alexander, Ing. Yi Min Shum, que fueron claves durante mi carrera, aquellos que ayudaban y enseñaban de la mejor manera posible siempre exigiendo para prepararnos para un futuro.

A compañeros que siempre estuvieron presente todo el tiempo como Esteban Giannattasio, Mauricio Cortes, Michelle Ramos y Arturo Manriquez con los que siempre podía contar para solventar problemas y quienes a pesar de todo me ayudaban de la forma que podían.

A mis familiares, sobre todo mi hermana por acompañarme a lo largo de la carrera.

A mis amigos Gabriel Cazola, María Pages, Miguel Silva, Argeo Lorenzo, Oriana Duarte, por apoyarme y estar siempre que los necesitaba.

Luis Alejandro González Rico

ÍNDICE GENERAL

INDICE GENERAL

CONTENIDO

ÍNDICE GENERAL	vii
INDICE DE FIGURAS.....	x
INDICE DE CUADROS.....	xii
RESUMEN.....	xiii
INTRODUCCIÓN	1

CAPÍTULO

I EL PROBLEMA	3
1.1. Planteamiento del Problema.....	3
1.2. Formulación del problema	4
1.3. Objetivos de la Investigación.....	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos	5
1.4. Justificación del problema	5
1.5. Alcance de la investigación.....	7
II MARCO TEÓRICO	9
2.1. Antecedentes	9
2.2. Bases Legales.....	10
2.3. Bases Teóricas.....	10
2.3.1. JavaScript.....	11
2.3.2. Node.js	12
2.3.3. Manejador de bases de datos	14
2.3.4. MongoDB	14
2.3.5. SQL.....	15

2.3.6.	NoSQL	16
2.3.7.	PHP	18
2.3.8.	Web Services	19
2.3.9.	Rest	20
2.3.10.	API	22
2.3.11.	Lenguaje Unificado de Modelado	24
2.3.12.	Cefalometría.....	25
2.4.	Definición de Términos Básicos	60
III MARCO METODOLÓGICO		64
3.1.	Tipo de Investigación.....	64
3.2.	Diseño de la Investigación	64
3.3.	Nivel de Investigación	64
3.4.	Población y Muestra.....	65
3.5.	Técnicas e instrumentos de recolección de información.....	65
3.6.	Fases Metodológicas	66
IV RESULTADOS		69
4.1.	Fase 1: Planificación del proyecto	69
4.1.1.	Historias de usuarios	69
4.1.1.1.	Aplicación de método de recolección de información.....	69
4.1.2.	Release Planning.....	72
4.1.3.	Iteraciones.....	73
4.1.4.	La velocidad del proyecto.....	73
4.1.5.	Programación en parejas.....	73
4.1.6.	Reuniones diarias.....	74
4.2.	Fase II – Diseño	74
4.2.1.	Diseños simples	74

4.2.1.1.	Diagrama de casos de uso.....	76
4.2.1.2.	Diagrama de Clases	78
4.2.1.3.	Diagrama de estados.....	80
4.2.2.	Glosario de términos.....	81
4.2.3.	Riesgos.....	87
4.2.4.	Funcionabilidad extra	88
4.2.5.	Refactorizar.....	88
4.3.	Fase III – Codificación.....	88
4.4.	Fase IV – Pruebas.....	88
V CONCLUSIONES Y RECOMENDACIONES		91
5.1.	Conclusiones	91
5.2.	Recomendaciones.....	93
BIBLIOGRAFÍA.....		94
ANEXOS		97

INDICE DE FIGURAS

Figura 1. <i>Modelo de trabajo de Node.js</i>	12
Figura 2. <i>Modelo de trabajo de Node.js con thread</i>	13
Figura 3. <i>Ángulo SNA</i>	27
Figura 4. <i>Ángulo SNB</i>	28
Figura 5. <i>Ángulo ANB</i>	29
Figura 6. <i>Ángulo SND</i>	30
Figura 7. <i>Segmento SL</i>	31
Figura 8. <i>Segmento SE</i>	32
Figura 9. <i>Ángulo Go-Gn S-N</i>	33
Figura 10. <i>Plano oclusal a S-N</i>	34
Figura 11. <i>Ángulo 1-NA</i>	35
Figura 12. <i>Distancia 1-NA</i>	36
Figura 13. <i>Ángulo Incisivo Superior - Plano Platino</i>	37
Figura 14. <i>Angulación del incisivo superior con SN</i>	38
Figura 15. <i>Ángulo 1 - NB</i>	39
Figura 16. <i>Distancia 1-NB</i>	40
Figura 17. <i>Incisivo Inferior al plano Go-Gn</i>	41
Figura 18. <i>Ángulo interincisal</i>	42
Figura 19. <i>Línea S</i>	43
Figura 20. <i>Convexidad Facial</i>	45
Figura 21. <i>Distancia A-Pg (Wits)</i>	46
Figura 22. <i>Eje Facial</i>	47
Figura 23. <i>Profundidad Facial</i>	48
Figura 24. <i>Ángulo del Plano Mandibular</i>	49
Figura 25. <i>Altura Facial Inferior</i>	50
Figura 26. <i>Arco Mandibular</i>	51
Figura 27. <i>Profundidad Maxilar</i>	52

Figura 28. Altura Maxilar	53
Figura 29. Incisivo Inferior 1 A -Pg (Plano dentario).....	54
Figura 30. Incisivo Inferior al Plano de Oclusión (1- Plano Oclusal).....	55
Figura 31. 1 Plano Mandibular (IMPA).....	56
Figura 32. 1 Plano Ba-N	57
Figura 33. Exposición del Incisivo Superior.....	58
Figura 34. Labio inferior con el plano E.....	59
Figura 35. Ángulo nasolabial inferior	60
Figura 36. Release Planning	73
Figura 37. Casos de uso del administrador	77
Figura 38. Casos de uso del usuario.....	78
Figura 39. Diagrama de Clases	79
Figura 40. Diagrama de estados.....	80

INDICE DE CUADROS

Tabla 1. Diferencia entre NoSQL y SQL.....	16
Tabla 2. Pruebas	68
Tabla 3. Glosario de términos	81
Tabla 4. Resultados de pruebas	89



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN**

**DESARROLLO DE SOFTWARE PARA TRAZADOS DE RADIOGRAFÍAS
ORTODÓNTICAS PARA LA FACULTAD DE ODONTOLOGÍA DE
LA UNIVERSIDAD JOSÉ ANTONIO PÁEZ FASE II**

Autor: González Rico Luis Alejandro

Tutor: MSc. Oneida Jiménez

Fecha: Junio 2017

RESUMEN

El presente trabajo de investigación tuvo como objetivo general desarrollar un software que realizara los cálculos necesarios para la obtención de ángulos y distancias específicas en los diferentes análisis cefalométricos, estos pueden ser la cefalometría de Steiner o la cefalometría de Ricketts, devolviendo los análisis requeridos por el usuario para poder entregar un análisis prudente al cliente. Para llevar a cabo este software se creó una API REST con el lenguaje de programación interpretado JavaScript y MongoDB como gestor de base de datos NoSQL, estos le permitieron a la API-REST dar resultados rápidos acorde a las respuestas que vaya recibiendo del usuario, para posteriormente dar un informe completo de todos los datos obtenidos. Esto se realizó a través de la metodología de desarrollo XP (programación extrema), ya que con sus diferentes fases se puede estudiar, analizar y diseñar un plan adecuado para la codificación de los diferentes algoritmos que se utilizarán para realizar las pruebas necesarias en un entorno controlado. Gracias al diseño de investigación especial que se utiliza se podrá realizar un estudio previo para establecer necesidades, funcionalidades y recursos necesarios para luego pasar al desarrollo del software propiamente. Entre las técnicas e instrumentos de recolección de información que se usaron están primeramente la de entrevista mixta para comenzar el levantamiento de información, y posteriormente la observación directamente a la muestra de la población que se estableció para la prueba del software.

Descriptor: análisis cefalométrico, software para trazado, programación extrema.

INTRODUCCIÓN

Un software cefalométrico es aquel software encargado de recibir, analizar y devolver los cálculos de las radiografías cefalométricas introducidas por los ortodoncistas ayudando a la interpretación de los resultados dando datos más precisos que los que se pueden obtener actualmente de manera manual. Las ventajas de estos softwares son su simplicidad a la hora de realizar cálculos complejos, solo necesitando la radiografía del paciente en cuestión y en poco tiempo dando un resultado altamente exacto.

En Venezuela estos softwares son prácticamente desconocidos, muy pocos profesionales son los que lo usan; y las instituciones que enseñan odontología no se pueden dar el lujo de comprar un software de análisis cefalométrico ya que es necesario el pago en dólares (\$), más el mantenimiento cada cierto tiempo para comprobar su buen funcionamiento a pesar de las circunstancias. En vista de esto se decidió desarrollar un software de trazado radiográfico para la Universidad José Antonio Páez que sirva a los estudiantes de dicha casa de estudio a practicar sus conocimientos aprendidos.

Para el desarrollo del software se dividió el sistema en dos (2) fases, lo que ayudaría con el factor tiempo y esfuerzo al ser más de un grupo de estudiantes el que está involucrado en el desarrollo de este software de trazado, realizando de forma armónica la parte de cliente-servidor que el sistema requiere para su buen funcionamiento. Este trabajo de grado se basará en el desarrollo del algoritmo que utilizará el software de trazado radiográfico para analizar los datos y dar un resultado preciso, se dividirá en 5 capítulos, cada uno con su función que se explicará a continuación:

Capítulo I: El problema, se expondrán las complicaciones que lleva la realización del proyecto, se describirán los objetivos de la investigación y la razón por el cual se está realizando este software.

Capítulo II: Marco Teórico, en este capítulo se presentarán las bases y fundamentos teóricos del proyecto, al igual que antecedentes de investigaciones de temas similares o que aporten información al trabajo de investigación.

Capítulo III: Marco Metodológico, se describirá el tipo, nivel y diseño de la investigación que caracterizarán el trabajo a realizar, de igual forma se elegirá una población y una muestra para realizar el estudio, así como también las técnicas utilizadas para la recolección de información. Y se describirá la metodología utilizada para el desarrollo que para este caso se utilizó XP por su facilidad y adaptabilidad al momento de ejecutar sus diferentes fases.

Capítulo IV: Resultados, se muestran los resultados que se obtuvieron en cada una de las fases de la metodología antes mencionada, se hacen usos de las técnicas de recolección de datos para determinar los requerimientos funcionales y no funcionales del sistema.

Capítulo V: Conclusiones y Recomendaciones, se hablará sobre las conclusiones llegadas, así como las recomendaciones propuestas.

CAPÍTULO I

EL PROBLEMA

1.1.Planteamiento del Problema

La ortodoncia es una especialidad de la odontología que se encarga del estudio, prevención, diagnóstico y tratamiento de las anomalías de forma, posición, relación y función de las estructuras dentomaxilofaciales. Su ejercicio es el arte de prevenir, diagnosticar y corregir sus posibles alteraciones y mantenerlas dentro de un estado óptimo de salud y armonía mediante el uso y control de diferentes tipos de fuerzas.

Las radiografías actuales son unas herramientas de diagnóstico fundamental para los odontólogos ya que permiten una detección rápida y eficaz de los posibles problemas que el paciente pueda tener en la cavidad bucal minimizando el margen de error en el diagnóstico médico y acortando los lapsos de espera para la obtención de resultados precisos. Existen diversos tipos de radiografías dentales con distintas finalidades y funciones donde el especialista es quien evalúa cuál tipo debe usar en una situación específica.

La radiografía panorámica le permite al clínico identificar el estado de las vías aéreas nasales, la simetría de la mandíbula y el estado general de la dentadura. En contraparte, la cefálica lateral ofrece la evaluación sagital de los huesos y establecer la severidad de la mala posición dental y esquelética. Esta severidad se determina mediante el cálculo de las distancias y ángulos entre varios sectores del cráneo que poseen rango de valores normales.

En los tiempos de ahora, este proceso se efectúa de forma manual, salvo centros especializados que tienen software de análisis cefalométricos especializados creados específicamente para su uso dentro de esas instalaciones. En las facultades de odontologías venezolanas se sigue manteniendo el método manual, ya que no se posee un capital lo suficientemente alto para invertir en un software de trazado comercial, lo que involucra más tiempo y la posible aparición de errores de cálculo por parte del

examinador, que utiliza estos datos para interpretar la complejidad del caso y sus interacciones.

Dado lo antes expresado se tiene la necesidad de crear un software para el trazado de estas radiografías adaptado a los requisitos de los usuarios, que además permita potenciar una odontología más ecológica al limitar el uso del papel para el trazado (papel cefalométrico) y el acetato de las radiografías. En consecuencia, y tomando en cuenta que este proyecto ha sido establecido en dos fases para su desarrollo, además de ser un proyecto de tipo especial con el objetivo de desarrollar la fase 2 que contempla el desarrollo de una API (Interfaz de programación de aplicaciones) para el cálculo de ángulos y distancias en función a alguno de los análisis cefalométricos seleccionados, permitiendo ingresar los puntos necesarios para exportar posteriormente una tabla con los resultados necesarios permitiéndole al odontólogo su correcta interpretación.

Entre las principales ventajas que presenta la creación de un software de trazado especial para la Universidad José Antonio Páez, es que se puede usar con fines educativos, permitiendo así un avance al momento de explicar a los estudiantes de la facultad de odontología el correcto método para éstos análisis, brindando un mejor servicio a los pacientes que se dirigen a las instalaciones en busca de un tratamiento.

Por último, teniendo un software propio se reducen gastos de frecuentes, como impresiones y materiales, sin mencionar que el margen de error que se puede producir es prácticamente nulo ya que es la API la encargada de realizar todos los cálculos necesarios en base a los puntos que los estudiantes en compañía del profesor ingresarían.

1.2. Formulación del problema

De acuerdo a lo antes expuesto existe una necesidad inmediata de crear e implementar en la facultad de odontología de la Universidad José Antonio Páez un software de trazado cefalométrico para así mejorar el nivel de aprendizaje de los estudiantes, y crear una base de datos que a futuro permita a los usuarios obtener más y mejores resultados en un tiempo menor al actual. Por ende, se formula la siguiente

interrogante ¿Cómo se pueden automatizar los cálculos manuales que realizan los especialistas odontólogos (ortodoncistas) para la obtención de resultados gracias a puntos cefalométricos previamente dados por imágenes radiográficas (cefálica lateral), donde se puedan extraer distancia y ángulos precisos que ayuden a los estudiantes de odontología de la Universidad José Antonio Páez?

1.3.Objetivos de la Investigación

1.3.1. Objetivo General

Desarrollar un software que realice los algoritmos para obtener los ángulos y distancias necesarios en una radiografía panorámica en la facultad de odontología de la Universidad José Antonio Páez.

1.3.2. Objetivos Específicos

- Analizar los resultados obtenidos en la Fase I del software para trazados de radiografías ortodónticas para la facultad de odontología de la Universidad José Antonio Páez determinando un modelo inicial para la Fase II.
- Iniciar la Fase II diseñando la API que permita realizar los cálculos de ángulos y distancias en función al trazado seleccionado y que posteriormente permita la exportación a una tabla con valores obtenidos.
- Elaborar la API según las pautas previamente establecidas para realizar los cálculos pertinentes a partir de los puntos obtenidos con las radiografías cefálicas.
- Probar la funcionalidad del software desarrollado dentro de un entorno controlado.

1.4.Justificación del problema

El presente trabajo de investigación se realiza con el fin de dar solución a un problema cotidiano que se presenta dentro de las instalaciones de la Universidad José Antonio Páez en la facultad de odontología, el cual es la realización de cálculos e interpretaciones de análisis cefalométricos de manera manual a radiografías (cefálica

lateral), lo cual puede ocasionar errores o demoras en ciertos tratamientos. Por ello se busca agilizar un proceso manual al convertirlo en automático, para así lograr que tanto los estudiantes como los profesores del campus tengan resultados más oportunos y acertados durante las prácticas realizadas.

Además, al convertir un proceso manual en automático, se eliminan ciertos factores que afectan los resultados finales, entre las ventajas que se obtienen al convertir el sistema se pueden mencionar:

- La mejor calidad del resultado obtenido: la reducción de costo ya que se reduce el tiempo para el análisis y entrega de resultados.
 - El dinero empleado para los análisis; ya que no se deben utilizar materiales no reutilizables.
 - Los tiempos de procesamiento de información, ya que todo se haría de manera inmediata para un análisis rápido.

Antes de realizar algún procedimiento a los pacientes de ortodoncia es necesario evaluar los elementos que forman la composición facial mediante un análisis cefalométrico, ya que mediante los diferentes puntos que toma, puede presentar un adecuado estudio del complejo craneomaxilofacial y dentario. La cefalometría en ortodoncia sirve como método auxiliar de diagnóstico de maloclusiones y malformaciones craneofaciales, su uso para la evaluación de los resultados no ha sido tan popular debido a que la mayoría de los análisis expresan sus valores de manera angular lo que no es de fácil interpretación para el odontólogo general.

Este análisis permite evaluar los cambios realizados a los pacientes, expresándolos en milímetros, con mediciones fáciles de realizar y de comprender, y demuestra la aplicabilidad en distintas disciplinas como la ortodoncia, ortodoncia maxilar, cirugía ortognática, medicina pediátrica, entre otras más. Un análisis cefalométrico entre sus principales aplicaciones se usa para:

- Estudio del crecimiento facial.

- Diagnóstico de posibles patologías especiales, tales como dilaceraciones, traumatismos hipertrofias adenoideas.
- Evaluación del espacio nasofaríngeo.
- Diagnóstico de anomalías cráneo-faciales (clase II, clase III).
- Seguimiento y evaluación de los resultados obtenidos en el tratamiento.

Actualmente los softwares que se encuentran en el mercado para la realización de cefalometría radiográficas no son muy usados, a pesar de ser potentes y capaces de proporcionar resultados muy preciosos y parecidos a los que se puede obtener de manera manual con los métodos convencionales. Es por ello que se desea crear un software de manera amigable entregando los resultados esperados de una manera clara y concisa para que así el ortodoncista sepa que tratamiento debe aplicar en un tiempo mucho menor y con un mejor control, ya que podría observar la evolución del tratamiento de una manera muy sencilla y cómoda tanto para él como para su paciente, sin el tedioso tiempo de análisis manual que éste requiere para evaluar un cambio o ver como evoluciona un tratamiento.

1.5. Alcance de la investigación

Tomando en consideración todo lo antes expuesto, el algoritmo desarrollado se encontrará en la API mencionada, que se mantendrá en el servidor para que el cliente creado en la Fase I se comuniquen con ella para enviar los puntos necesarios de las cefalometrías y recibir los resultados del análisis indicado. Ésta será una de las partes que integrará todo el software para trazados de radiografías ortodónticas, ya que por limitantes de tiempo y complejidad del propio software no es posible realizarlo como software completo, sino que es necesario la división en módulos. En el caso de este trabajo de grado se abarcará la Fase II del proyecto, la cual comprende la realización del algoritmo que permita la interpretación de los puntos ingresados gracias a las imágenes de las radiografías cefalométricas para dar los cálculos solicitados por los usuarios de una manera sencilla y entendible, ahorrando así tiempo que puede ser utilizado en otra labor.

La metodología a utilizar en este proyecto será la metodología XP (Programación Extrema), la cual es un procedimiento rápido y ágil que nos permitirá aplicar las mejores técnicas de desarrollo durante el ciclo de vida del software de una manera dinámica para así obtener los mejores resultados.

El tiempo que se dispone para el desarrollo de dicho proyecto consta de ocho (8) meses, y el lugar donde se desarrollará es en la Universidad José Antonio Páez (UJAP), que se ubica en el municipio San Diego estado Carabobo.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes

Se tomaron en consideración una serie de trabajos de grado realizados previamente por diferentes autores que nos servirán de guía durante el desarrollo de este trabajo de grado. Estos trabajos se describen a continuación de manera breve.

Sáez Martínez, Boronat Catalá, Muñoz Castro, Zamora Martínez, Tarazona Álvarez, Gangía Franco (2015), Comparative study between cephalometric superposition analysis drawn digitally and manually with Dolphin® and Nemoceph®: Are there differences?, Universidad de Valencia, España. El estudio realizado dentro de la Universidad de Valencia permitía comparar los resultados obtenidos de manera manual y digital con 2(dos) softwares de cefalometría, estudiando a los mismos pacientes bajo las mismas condiciones, se concluyó que el método manual mostraba resultados casi perfectos muy parecidos a uno de los softwares utilizados, mientras que el segundo software presentaba muchas variaciones entre los resultados iniciales y finales. Este estudio comparativo nos permite ver como se ha logrado obtener resultados de manera automática a través de un software, ahorrando tiempo y dinero y con resultados bastante parecidos a los que se obtienen de manera habitual con los métodos manuales actuales.

Guevara (febrero, 2013) Desarrollo de un sistema de información bajo plataforma web para el control de facturación e inventario en la empresa representaciones Willedu C.A. En este trabajo de grado se puede ver el hincapié en los modelos de lenguaje unificados (UML) basados en la metodología de desarrollo Programación Extrema (XP) que aportan una correcta recolección y análisis de los requerimientos funcionales y no funcionales para el completo desarrollo del sistema de información.

Morales, Orellana, Garcia, Ramírez R. y Setien D. (2007) La hipermedia y la enseñanza-aprendizaje de odontología: proyecto factible empleando el software recompx®, Universidad de los Andes, Venezuela. El propósito de esta investigación fue comprobar que tan factible es el uso de softwares multimedia para los estudiantes de odontología de la Universidad de los Andes para la adquisición de conocimientos, el dominio de contenidos conceptuales y procedimentales para lograr un correcto desarrollo de las habilidades y destrezas necesarias en una práctica clínica odontológica eficiente. Con él se pudo comprobar la eficiencia de estos softwares ya que los resultados que se obtuvieron favorecen el aprendizaje de conocimientos odontológicos, agilizando el proceso de aprendizaje gracias a un nuevo método más moderno, dinámico e interactivo con el estudiante.

2.2.Bases Legales

Según Villafranca D. (2002) “Las bases legales no son más que leyes que sustentan de forma legal el desarrollo del proyecto” explica que las bases legales “son leyes, reglamento y normas necesarias en algunas investigaciones cuyo tema así lo amerite”.

Para el desarrollo de esta API se va a usar una licencia de copyleft de tipo GNU GPL v3, que permite el uso de la API en modo comercial, para su distribución y modificación, además de que se puede usar de manera privada y usar las patentes. Siempre y cuando se cumplan las siguientes condiciones: que se mantenga la API de código abierto, bajo la misma licencia, incluyendo una copia de la licencia junto al código de la API e indicando los cambios realizados dentro del código. Pero la licencia hace que el autor se libere de las responsabilidades del código especificando que él no da garantías del código ni es responsable del sistema donde se use.

2.3.Bases Teóricas

Las bases teóricas construyen el corazón del trabajo de investigación, pues es sobre estas bases que se construye todo el trabajo. Estas ayudaran en la estructura sobre la cual se diseñará el estudio, sin estas no se sabe cuáles elementos se pueden tomar en

cuenta y cuáles no. A continuación se presentan las bases teóricas que sustentan la investigación sobre el uso de las tecnologías utilizadas y la información necesaria para el correcto desarrollo del Software de Trazado.

2.3.1. JavaScript

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en el ámbito de los nuevos lenguajes de programación web.

JavaScript (“JS” para abreviar) es un lenguaje de programación dinámico completo que, cuando se aplica a un documento HTML, puede proporcionar interactividad dinámica en sitios web. JavaScript es increíblemente versátil. Puede comenzar con pequeños, con carruseles, galerías de imágenes, diseños fluctuantes y respuestas a los clicks del botón. Con más experiencia podrás crear juegos, gráficos animados en 2D y 3D, aplicaciones completas basadas en bases de datos y mucho más.

JavaScript en si es bastante compacto pero muy flexible. Los desarrolladores han escrito una gran variedad de herramientas que complementan el lenguaje JavaScript principal, desbloqueando una gran cantidad de funcionalidad adicional con un mínimo esfuerzo. Estas incluyen:

- Interfaces de programación de aplicaciones (Application Programming Interfaces, API) integradas en los navegadores web, que proporcionan funcionalidad como la dinámica de HTML y la configuración de estilos CSS, la recopilación y manipulación de un flujo de video desde la cámara web del usuario o la generación de gráficos 3D y muestras de audio.
- API de terceros que permiten a los desarrolladores incorporar funcionalidades en sus sitios de otros proveedores de contenidos, como lo son Twitter o Facebook.
- Los frameworks y bibliotecas de terceros que puede aplicar a su HTML para permitirle crear rápidamente sitios y aplicaciones.

JavaScript no está limitado sólo a los navegadores web, en realidad puede correr en cualquier ambiente que tenga instalado el intérprete para sus instrucciones (también conocido como motor). Esta flexibilidad ha hecho que a partir del año 2000 haya una creciente proliferación de implementaciones de JavaScript en el lado del servidor, siendo Node.js una de las más populares. Además, JavaScript es un lenguaje muy

poderoso y tiene una fuerte inclinación hacia paradigmas funcionales, esto lo hace muy atractivo para algunos desarrolladores.

2.3.2. Node.js

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo. Su diseño está enfocado en maximizar el rendimiento y la eficiencia. El sistema de comunicación no bloqueante y los eventos asíncronos lo convierte en una opción ideal para el desarrollo de aplicaciones que manejan datos en tiempo real o de cómputo intensivo. Es decir, Node.js es un framework que trabaja sobre JavaScript que te proporciona herramientas para que puedas construir desde aplicaciones web hasta programas de cómputo intensivo.

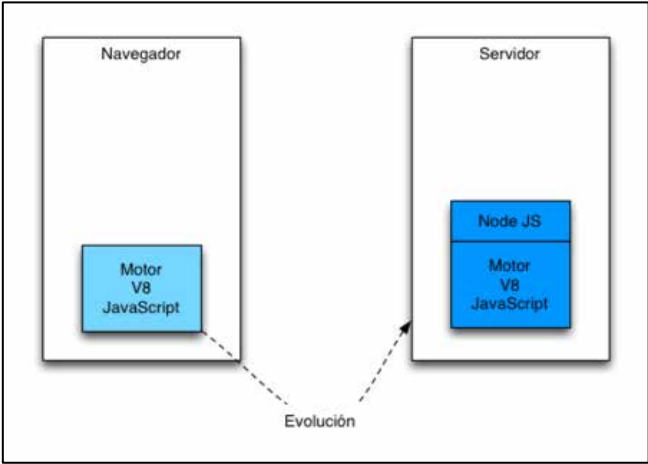
El motor V8 de Google

Node.js está basado en el motor V8 de JavaScript de Google. Este motor está diseñado para correr en un navegador y ejecutar el código de JavaScript de una forma extremadamente rápida. La tecnología que está detrás de Node.js permite ejecutar este motor en el lado del servidor abriendo un nuevo abanico de posibilidades en cuanto al mundo de desarrollo se refiere. Para ello el entorno de Node.js ha desarrollado un conjunto de amplias librerías que no tienen nada que envidiar a otras plataformas. Por otro lado, se han eliminado algunas funcionalidades que en el entorno de servidor no tenían sentido como por ejemplo el uso de Document Object Model.

Threads(Hilos)

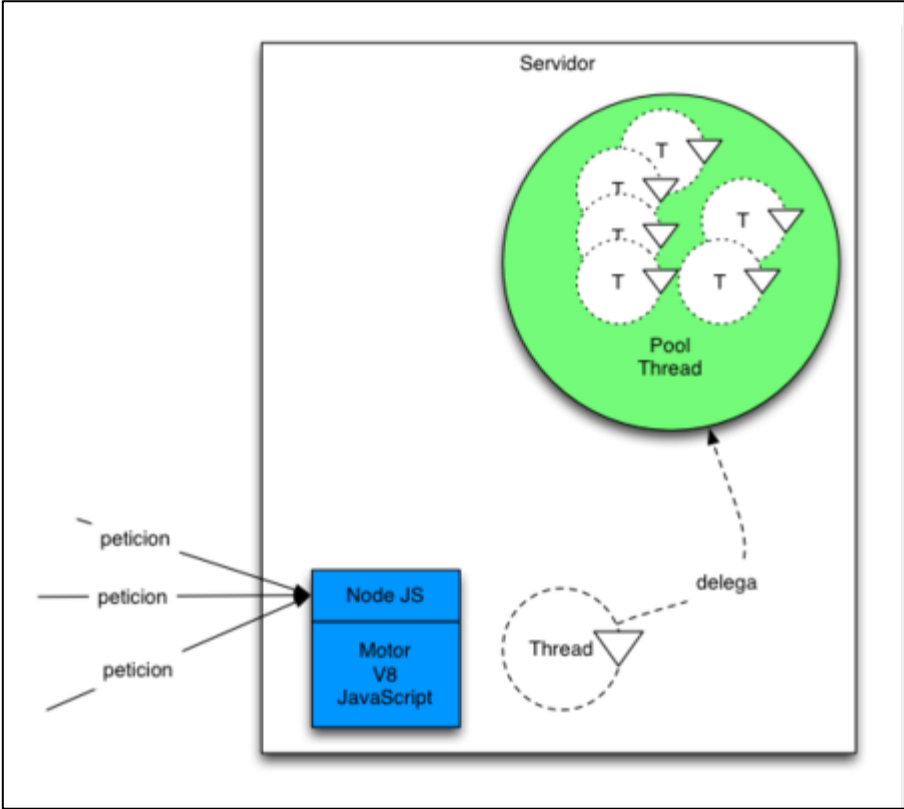
Un Thread es un mecanismo que permite a una aplicación realizar varias tareas a la vez de manera concurrente. Los hilos de ejecución que comparten los mismos recursos, sumados a estos recursos, son en conjunto conocidos como un proceso. El hecho de que los hilos de ejecución de un mismo proceso compartan los recursos hace que cualquiera de estos hilos pueda modificar estos recursos. Cuando un hilo modifica un dato en la memoria, los otros hilos acceden a ese dato modificado inmediatamente. Node.js trabaja con un único hilo de ejecución que es el encargado de organizar todo el flujo de trabajo que se deba realizar, todo realizándolo de forma asíncrona, esto permite trabajar de una forma óptima ya que delega todo el trabajo en un pool de threads (una piscina de hilos). Este pool de threads está construido con la librería libuv y es a donde Node.js envía el trabajo que hay que realizar al pool.

Figura 1. *Modelo de trabajo de Node.js*



Fuente: Álvarez Cecilio (2014)

Figura 2. Modelo de trabajo de Node.js con thread



Fuente: Álvarez Cecilio (2014)

2.3.3. Manejador de bases de datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. (Pérez D., 2007)

Características de una Base de Datos

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoria.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

2.3.4. MongoDB

MongoDB es la base de datos NoSQL líder, al ser una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta.

MongoDB ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidados. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria. La replicación nativa de MongoDB y la tolerancia a los fallos automática ofrece fiabilidad a nivel empresarial.

Casos de uso de MongoDB:

La base de datos de MongoDB es adecuada para los siguientes usos:

- Almacenamiento y registros de eventos.
- Para sistemas de manejo de documentos y contenido.
- Comercio electrónico.

- Juegos.
- Problemas de alto volumen de lecturas.
- Aplicaciones móviles.
- Almacén de datos operacionales de una página web.
- Manejo de contenido.
- Almacenamiento de comentarios.
 - Votaciones.
 - Registro de usuarios
 - Perfiles de usuarios
 - Sesiones de datos
- Proyectos que utilizan metodologías de desarrollo iterativos o ágiles.
- Manejo de estadísticas en tiempo real

MongoDB es utilizado para uno o varios de estos casos por varias empresas.

2.3.5. SQL

SQL (Lenguaje de Consulta Estructurado) es un lenguaje estándar para acceder y manipular bases de datos, y está diseñado bajo el estándar ANSI (American National Standards Institute).

RDBMS

RDBMS significa Relational Database Management System, es la base para SQL, y todos los sistemas de bases de datos modernos, tales como MS SQL Server, IBM DB2, Oracle, MySQL, Microsoft Access. Los datos de RDBMS se almacenan en objetos de base denominadas tablas. Una tabla es una colección de entradas de datos relacionadas, y estos consiste en columnas y filas. Cada tabla se divide en las entidades más pequeñas llamadas campos. Un campo es una columna de una tabla que está diseñada para mantener información específica sobre cada registro de la tabla. Un registro también es llamado fila, es cada entrada individual que existe. Una columna es una entidad vertical en una tabla que contiene toda la información asociada a un campo específico en una tabla.

2.3.6. NoSQL

Originalmente el término NoSQL (1998) se refería a una base de datos relacional de código abierto que no usaba un lenguaje SQL (Structured Query Language). Y no fue hasta el 2009 que comenzaron a tratar a NoSQL como una base de datos distribuida de código abierto no relacional (Not-Only SQL).

Las características comunes entre todas las implementaciones de bases de datos distribuidas no relacionales, propietarias o no, suelen ser las siguientes:

- **Consistencia Eventual:** No se implementan mecanismos rígidos de consistencia como los presentes en las bases de datos relacionales, donde la confirmación de un cambio implica una comunicación del mismo a todos los nodos que lo repliquen. Esta flexibilidad hace que la consistencia se dé, eventualmente, cuando no se hayan modificados los datos durante un periodo de tiempo. Esto se conoce como BASE (Basically Available Soft-state Eventual Consistency, o coherencia eventual flexible básicamente disponible), en contraposición a ACID, su analogía en las bases de datos relaciones.
- **Estructura distribuida:** Generalmente se distribuyen los datos mediante mecanismos de tablas de hash distribuidas (DHT) ya que realmente se trata, según las distintas implementaciones, de redes p2p.
- **Escalabilidad horizontal:** La implementación típica se realiza en mudos nodos de capacidad de procesado limitado, en vez de utilizar grandes Mainframes.
- **Tolerancia a fallos y Redundancia.**

Para entender mejor las diferencias que existen entre SQL y NoSQL ver la siguiente tabla (ver Tabla 1).

Tabla 1. Diferencia entre NoSQL y SQL

	NoSQL	SQL
	No-relacional	Relacional
Modelo	Almacena datos en JSON documents, en pares de llaves / valores, con almacenes de columnas o gráficos	Almacena datos en tablas

	NoSQL	SQL
Datos	Ofrece flexibilidad ya que no todos los registros deben almacenar las mismas propiedades	Ideal para soluciones donde cada registro tiene las mismas propiedades
	Nuevas propiedades pueden añadirse sobre la marcha	La adición de una nueva propiedad puede requerir la modificación de esquemas o el relleno de datos
	Las relaciones se capturan a menudo desnormalizando datos y presentando todos los datos en un objeto en un solo registro	Las relaciones se capturan a menudo en el modelo normalizado usando juntas para resolver referencias a través de tablas
	Buena para datos semi-estructurados, complejos o anidados	Buena para datos estructurados
Schemas	Esquemas dinámicos o flexibles	Esquema estricto
	La base de datos es esquema-agnóstica y el esquema es dictado por la aplicación. Esto permite agilidad y desarrollo altamente iterativo	El esquema debe mantenerse y quedarse sincronizado entre la aplicación y la base de datos
Transacciones	El soporte de transacciones ACID varía según la solución	Soporte de transacciones ACID
Consistencia y disponibilidad	Consistencia eventual a fuertemente apoyada, dependiendo de la solución	Fuerte consistencia aplicada

	NoSQL	SQL
	La consistencia la disponibilidad y el desempeño se pueden negociar para satisfacer las necesidades de la aplicación (Teorema CAP)	Se prioriza la coherencia sobre la disponibilidad y el rendimiento
Rendimiento	El rendimiento se puede maximizar reduciendo la consistencia, si es necesario	Insertar y actualizar el rendimiento depende de lo rápido que se cometa una escritura, ya que se aplica una consistencia fuerte. El rendimiento se puede maximizar mediante la ampliación de los recursos disponibles y el uso de estructuras en memoria
	Toda la información sobre una entidad normalmente se encuentra en un solo registro, por lo que una actualización puede ocurrir en una operación	La información y la entidad pueden estar distribuidas entre muchas tablas o filas, requiriendo que muchas combinaciones se completen y actualicen, o una consulta
Escala	Normalmente, la escala se consigue horizontalmente con datos divididos para extender servidores	Normalmente, la escala se consigue verticalmente con más recursos del servidor

Fuente: Microsoft (2017).

2.3.7. PHP

PHP (acrónimo recursivo de PHP: Hypertext Proprocessor) Es un lenguaje de código abierto muy popular especialmente adecuado al desarrollo web y que puede ser

incrustado en HTML. Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas cosas y se pueden mencionar las siguientes:

- **Scripts del lado del servidor:** Este es el campo más tradicional y el foco principal. Son necesarias 3 cosas para que funcione, estas son: Analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web.
- **Scripts desde la línea de comando:** Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador PHP para utilizarlo de esta manera.
- **Escribir aplicaciones de escritorio:** Se puede utilizar PHP-GTK para escribir aplicaciones de escritorio con una interfaz gráfica de usuario.

2.3.8. Web Services

Un Web Service o servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet. Entre los estándares más empleados tenemos:

- **Web Service Protocol Stack:** conjunto de servicios y protocolos de los servicios web.
- **XML (Extensible Markup Language):** formato estándar para los datos que se vayan a intercambiar.
- **SOAP (Simple Object Access Protocol):** protocolo sobre el que se establece el intercambio.

- **WSDL (Web Services Description Language):** es el lenguaje de la interfaz pública para los servicios web. En una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios.
- **UDDI (Universal Description, Discover and Integration):** protocolo para publicar la información de los servicios web. Permite comprobar que servicios web están disponibles.
- **WS-Security (Web Service Security):** protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Este garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.
- **REST (Representational State Transfer):** arquitectura que, haciendo uso del protocolo HTTP, proporciona una API que utiliza cada uno de sus métodos (GET, POST, PUT, DELETE, etc.) para poder realizar diferentes operaciones entre la aplicación que ofrece el servicio web y el cliente.
- **GraphQL:** una arquitectura alternativa a REST.

2.3.9. Rest

REST (REpresentational State Transfer) es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC. REST se definió en el 2000 por Roy Fielding, podríamos considerar a REST como un framework para construir aplicaciones web respetando HTTP.

Por lo tanto, REST es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet. Existen tres niveles de calidad a la hora de aplicar REST en el desarrollo de una aplicación web y más concretamente una API que se recogen en un modelo llamado Richardson Maturity Model en honor al hombre que lo estableció, Leonar Richardson, padre de la arquitectura orientada a recursos. Esos niveles son:

· Nivel 1: Uso correcto de URIs

Cuando desarrollamos una web o una aplicación web, las URLs nos permiten acceder a cada una de las páginas, secciones o documentos del sitio web. Cada página, información en una sección, archivo, cuando hablamos de REST, los nombramos como recursos. El recurso por lo tanto es la información a la que queremos acceder o que queremos modificar o borrar (independientemente de su formato). Las URL (Uniform Resource Locator) son un tipo de URI (Uniform Resource Identifier) que además de permitir identificar de forma única el recurso, nos permite localizarlo para poder acceder a él o compartir su ubicación.

Una URL se estructura de la siguiente forma:

{protocolo}:// {dominio o hostname} [: puerto (opcional)]/ {ruta del recurso}? {consulta de filtrado}

Existen varias reglas básicas para ponerle nombre a las URI de un recurso:

- Los nombres de URI no deben implicar acción, por lo tanto, debe evitarse usar verbos en ellos.
- Deben ser únicas, no debemos tener más de una URI para identificar un mismo recurso.
- Deben ser independiente de formato.
- Deben mantener una jerarquía lógica.
- Los filtrados de un recurso no se hacen en la URI.

· Nivel 2: HTTP

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente (se le suele llamar “agente de usuario”) realiza una petición enviando un mensaje, con cierto formato al servidor. El servidor (se le suele llamar servidor web) le envía un mensaje de respuesta. Ejemplos de clientes son los navegadores web.

Para desarrollar una API-REST los aspectos claves que hay que dominar y tener claros son:

Métodos de HTTP

HTTP define una serie predefinida de métodos de petición (algunas veces llamados verbos) que pueden utilizarse. Cada método indica la acción que desea que se efectúe sobre el recurso identificado. Lo que este recurso representa depende de la aplicación del servidor. Entre los más importantes para tener en cuenta tenemos:

- **GET:** Para consultar y leer recursos.
- **POST:** Para crear nuevos recursos.
- **PUT:** Para editar recursos ya existentes.
- **DELETE:** Para eliminar un recurso.
- **PATCH:** Para editar partes concretas de un recurso.

Códigos de respuesta

El código de respuesta o retorno es un número que indica que ha pasado con la petición. El resto del contenido de la respuesta dependerá del valor de este código. Cada código tiene un significado concreto. Sin embargo, el número de los códigos están elegidos para que se pueda identificar el tipo de respuesta que ha dado el servidor:

- **Códigos con formato 1xx:** Respuestas informativas. Indican que la petición ha sido recibida y se está procesando.
- **Códigos con formato 2xx:** Respuestas correctas. Indica que la petición ha sido procesada correctamente
- **Códigos con formato 3xx:** Respuestas de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.
- **Códigos con formato 4xx:** Errores causado por el cliente: Indica que ha habido un error en el procesado de la petición a causa de que el cliente ha hecho algo mal.
- **Códigos con formato 5xx:** Errores causado por el servidor. Indica que ha habido un error en el procesado de la petición a causa de un fallo en el servidor.

· **Nivel 3: Hypermedia**

Significa conectar mediante vínculos las aplicaciones clientes con las APIs, permitiendo a dichos clientes despreocuparse por conocer de antemano del cómo acceder a los recursos. Con Hypermedia básicamente añadimos información extra al recurso sobre su conexión a otros recursos relacionados con él.

2.3.10. API

Una API es el mecanismo más útil para conectar dos softwares entre sí para el intercambio de mensajes o datos en formato estándar como XML o JSON. Igual que una interfaz de usuario permite la interacción y comunicación entre un software y una persona, una API (Application Programming Interface) facilita la relación entre dos aplicaciones para el intercambio de mensajes o datos. Un conjunto de funciones y

procedimientos que ofrece una biblioteca para que otro software la utilice como capa de abstracción, un espacio de acceso e intercambio de información adicional en la parte superior. Así una se sirve de la información de la otra sin dejar de ser independientes.

Cada API está diseñada en un lenguaje de programación concreto y con unas especificaciones distintas que la definen (estas pueden incluir especificaciones para estructuras de datos y rutinas, clases de objetos o variables, a partir de las cuales se basa el uso de esa interfaz)

Tipos de APIs

- **APIs de servicio web:**

Son las interfaces de desarrollo de aplicaciones que permiten el intercambio entre un servicio web (software que da acceso a un servicio concreto a través de una URL) y una aplicación. Normalmente ese intercambio se produce a través de peticiones HTTP o HTTPS (la versión cifrada del protocolo HTTP). En la petición de la aplicación y respuesta, también en HTTP del servicio web, se contiene información de todo tipo tanto en los metadatos de la cabecera como en los del mensaje, normalmente en dos tipos de formatos muy usados: XML o JSON.

Hay 4 tipos de API de servicio web habituales:

- **SOAP (Simple Object Access Protocol):** un protocolo estándar de intercambio de información y datos en XML entre dos objetos.
- **XML-RPC:** un protocolo de llamada de procedimiento remoto que usa XML como formato de datos y llamadas HTTP como sistema de comunicación.
- **JSON-RPC:** mismo protocolo que el anterior, pero con formato JSON
- **REST:** Arquitectura de software para sistemas hipermedia en la World Wide Web.

- **APIs basadas en bibliotecas:**

Este tipo de APIs son las que permiten que una aplicación importe una biblioteca de otro software para hacer el intercambio de información. Hoy en día gran parte de las bibliotecas que dan acceso a productos y servicios están diseñadas en JavaScript. Las APIs en JavaScript suelen ser un ejemplo ilustrativo de APIs basadas en bibliotecas, por ejemplo, las que se utilizan dentro del mercado de la cartografía web (servicios como Google Maps, Leaflet, ArcGIS, CartoDB, MapBox o D3.js).

- **APIs basadas en clases:**

Este tipo de interfaces de desarrollo de aplicaciones permite la conexión con los datos en torno a las clases, como es habitual en programación orientada a objetos con Java. La API de Java usa clases abstractas para la creación de aplicaciones igual que cualquier programa desarrollado en este lenguaje. Esas clases proporcionan todo lo necesario para realizar todo tipo de funciones dentro de esas aplicaciones. La interfaz de desarrollo de Java se organiza en paquetes y cada uno de esos paquetes contiene a su vez un conjunto de clases relacionadas entre sí.

- **APIs de funciones en sistemas operativos:**

Sistemas operativos como Windows disponen de APIs que permiten esa comunicación entre programas y el OS. Entre las APIs se pueden mencionar: Interfaz de usuario, acceso y almacenamiento de datos, mensajería, gráficos y multimedia, diagnóstico de errores, etc.

2.3.11. Lenguaje Unificado de Modelado

(UML, Unified Modeling Language) es el lenguaje de modelado de sistemas de sistemas de software más conocido y utilizado en la actualidad; aun cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje Gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguaje de programación, esquemas de base de datos y componentes de software reutilizables. (uml.org). Los diferentes tipos de diagramas UML según Muller:

- **Diagramas de clases:** Representa la estructura estática de un sistema en términos de clases y relaciones entre estas clases. Un diagrama de clases no expresa nada en particular sobre los enlaces de un objeto dado, pero describe de manera abstracta los enlaces potenciales de un objeto hacia otro objeto.
- **Diagrama de casos de uso:** Los casos de uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario; permiten definir los límites de un sistema de relaciones entre un sistema y el entorno.
- **Diagramas de objetos:** Los diagramas de objetos o diagramas de instancias, muestran objeto y clases. Como los diagramas de clases los diagramas de objeto muestran las estructuras estáticas. Son utilizados principalmente para mostrar un contexto, pero también para facilitar la comprensión de las estructuras de datos complejas, como las estructuras recursivas.

- **Diagrama de Secuencias:** Estos diagramas muestran interacciones entre los objetos insistiendo en la cronología de los envíos de mensajes. Un objeto se materializa por un rectángulo y una barra vertical llamada línea de vida de los objetos.
- **Diagrama de Colaboración:** Muestran interacciones entre objetos, insistiendo más particularmente en la estructura especial estática que permite la colaboración de un grupo de objetos.
- **Diagrama de Estados:** Los diagramas de estados o transiciones visualizan autómatas de estados finitos, desde el punto de vista de los estados y transiciones.
- **Diagrama de Actividades:** Un diagrama de actividades es una variante de los diagramas de estado, organizado respecto a las acciones y principalmente destinado a representar el comportamiento interno de un método o de un caso de uso.
- **Diagrama de Componentes:** Los diagramas de componentes describen los elementos físicos y sus relaciones en el entorno de realización. Los diagramas de componentes muestran las operaciones de realización.
- **Diagramas de despliegue:** Estos diagramas muestran la disposición física de los distintos materiales que entran en la composición de un sistema y el reparto de los programas ejecutables sobre estos materiales.

2.3.12. Cefalometría

La cefalometría se define como el conjunto de mediciones, que sobre radiografías cefálicas de frente o de perfil, hacemos utilizando una serie de puntos, líneas, planos y ángulos pre-establecidos por investigadores que realizaron y publicaron sus respectivos análisis cefalométricos. La cefalometría desde su aparición, se convirtió en uno de los medios más utilizados por el ortodoncista para realizar un examen clínico completo, un correcto diagnóstico y una adecuada planificación para el tratamiento de las diversas displasias dento-esqueleto-funcionales.

Los problemas de maloclusión pueden presentar un origen real de tipo esquelético, acompañado de problemas de tipo dentario, de allí la importancia de la cefalometría en el análisis de las maloclusiones. Algunos casos de maloclusiones son

los de clase I (apiñamiento, mordida cruzada anteriores), clase II (prognatismo), clase III (retrognatismo), maloclusiones transversales (mordidas cruzadas unilaterales, mordidas cruzadas bilaterales, mordidas en tijera, signo de brody), y maloclusiones verticales (supermordida, mordida borde a borde, mordida abierta).

Cefalometría de Steiner

Publicada a principios de los años (¿) por Cecil C. Steiner en el American Journal of Orthodontics con el título "Cephalometrics for you and me". Para su desarrollo Steiner partió de otros análisis cefalométricos ya existentes como son los análisis cefalométricos de Downs y Wylie. El análisis cefalométrico de Steiner utiliza como plano de referencia el Plano Silla-Nasion (S.N). El Análisis de Steiner fué un elemento de gran importancia en el ámbito científico para el diagnóstico en ortodoncia. Hoy en día sigue siendo utilizado, siendo uno de los preferidos dentro de la comunidad de ortodoncistas.

Áreas de estudio del análisis cefalométrico de Steiner son:

- Análisis esquelético
- Análisis dental
- Análisis de tejidos blandos

Los ángulos y segmentos que se evalúan durante el análisis cefalométrico de Steiner son:

- **Posición Anteroposterior de la Maxila**

Ángulo SNA: Formado por los planos SN (Silla – Nasion) y NA (Nasion – Punto A). Ver **Figura 3**.

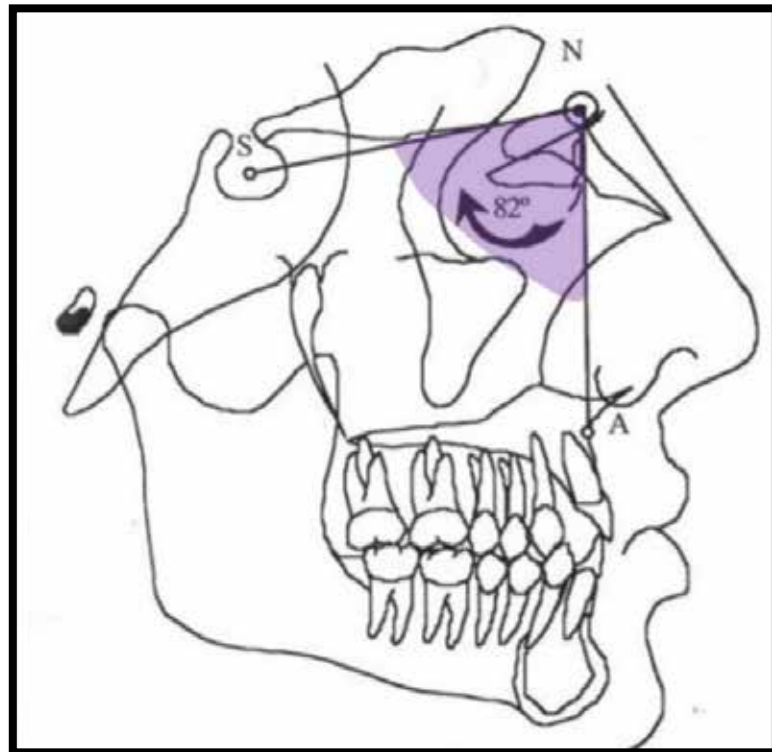
Norma: 82°.

Interpretación:

- Este ángulo indica la ubicación anteroposterior de la maxila con respecto a la base del cráneo.

- Mayor a 82° : Maxila protrusiva en relación a la base del cráneo.
- Menos a 82° : Maxilar retruido.
- Se debe tener en cuenta que si la línea SN se encuentra muy horizontal o muy verticalizada la medida puede ser modificada
- Así mismo si el punto N se encuentra más hacia adelante o más hacia atrás.

Figura 3. Ángulo SNA



Fuente: Parejas Freddy (2013)

· **Posición Anteroposterior de la Mandíbula**

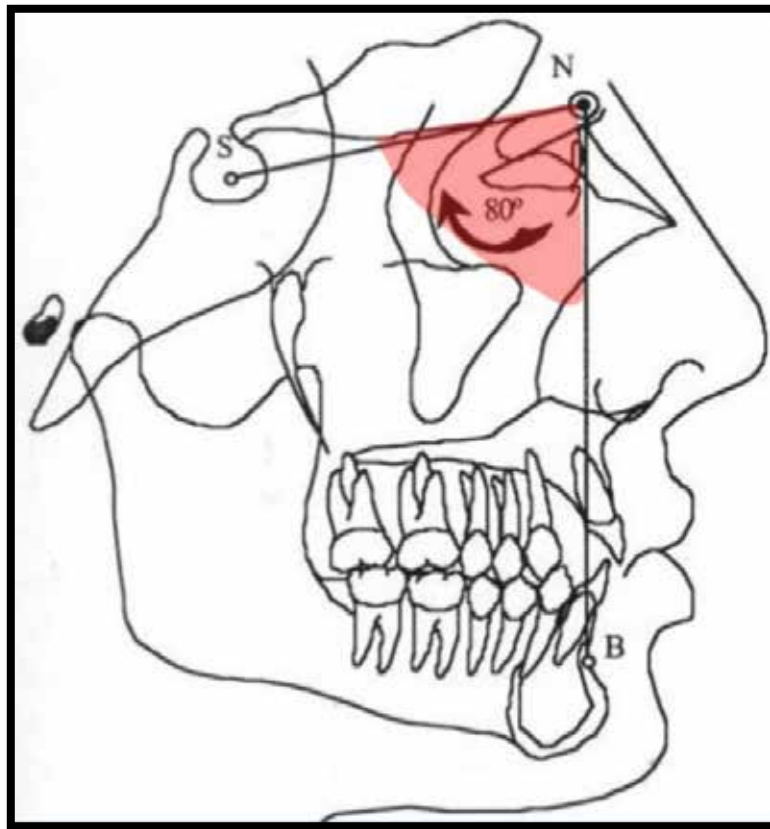
Ángulo SNB: Formado por los planos SN (Silla – Nasion) y NB (Nasion – Punto B). Ver **Figura 4.**

Norma: 80° .

Interpretación:

- Este ángulo indica la ubicación anteroposterior de la mandíbula con respecto a la base del cráneo.
- Mayor a 80° : Mandíbula protruida.
- Menos a 80° : Mandíbula retruida.
- Cuando se presenta alguna alteración en la inclinación y la longitud de la base del cráneo debido a la ubicación de Nasion. El ángulo se puede abrir o cerrar ligeramente.

Figura 4. *Ángulo SNB*



Fuente: Parejas Freddy (2013)

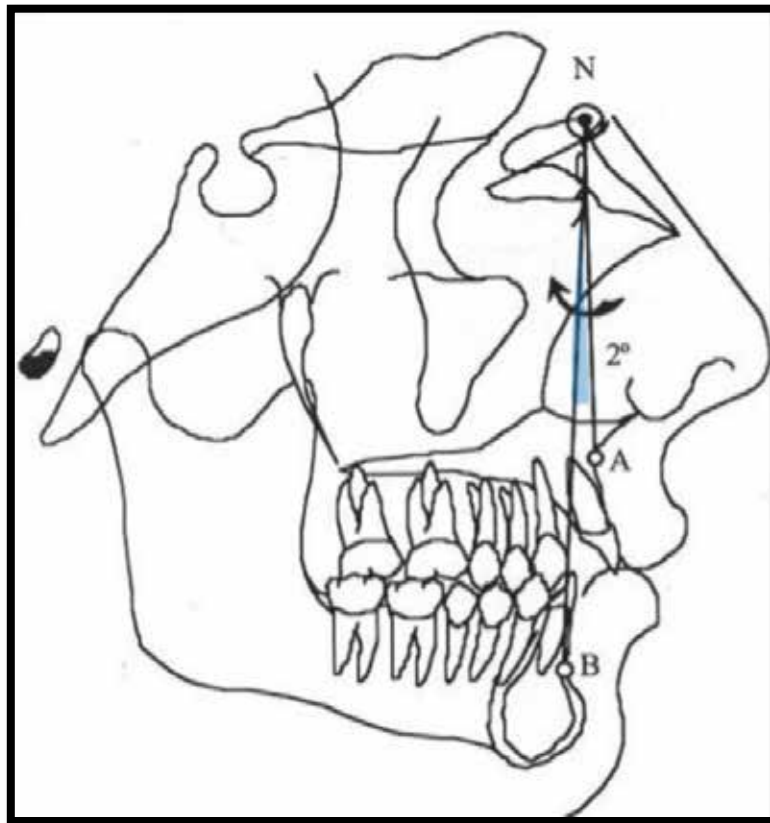
Ángulo ANB: Es el ángulo formado por los planos NA (Nasion – Punto A) y NB (Nasion – Punto B). Ver **Figura 5**.

Norma: 2°.

Interpretación:

- Indica la relación anteroposterior que existe entre la maxila y la mandíbula.
- Los ángulos aumentados indican relación clase II.
- Los ángulos negativos cuando Plano N – B por delante del plano N – A, indican una relación clase III.
- El ángulo indica la relación maxilo-mandibular, pero no indica si el problema se debe al maxilar o a la mandíbula.
- Si hay base craneal corta, el ángulo se puede abrir.
- Si hay base craneal larga, el ángulo se cierra.

Figura 5. Ángulo ANB



Fuente: Parejas Freddy (2013)

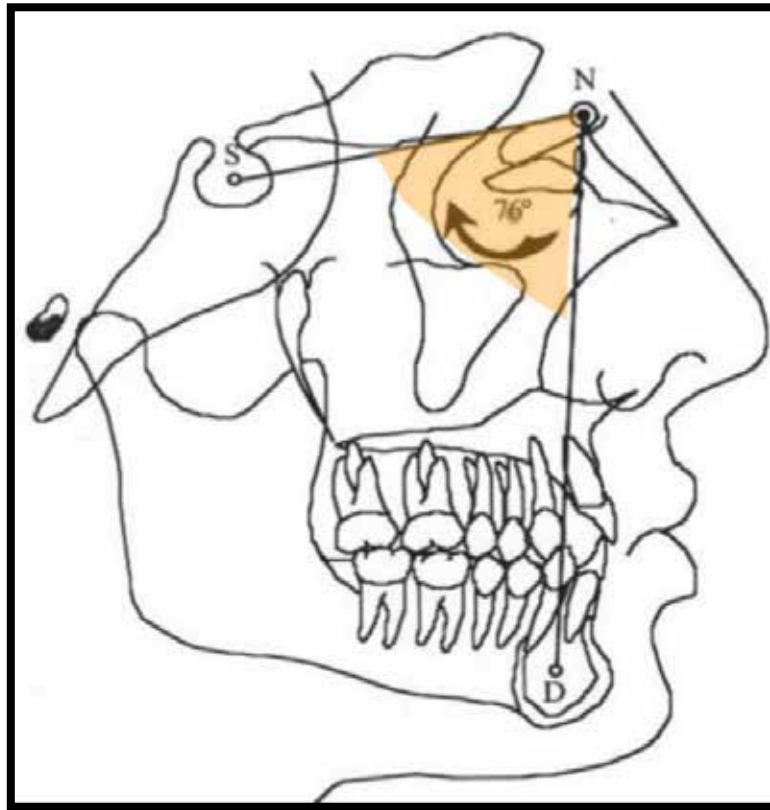
Ángulo SND: Es el ángulo formado por los planos SN (Silla – Nasion) y ND (Nasion – Punto D). Ver **Figura 6**.

Norma: 76°.

Interpretación:

- Cómo el punto D representa el centro de la sínfisis, este ángulo indica la ubicación anteroposterior basal de la mandíbula con respecto a la base del cráneo.
- Esta medida se toma para complementar la lectura SNB.
- Un ángulo mayor a 76° indica una mandíbula protruida.
- Un ángulo menos a 76° indica una mandíbula retruida.

Figura 6. Ángulo SND



Fuente: Parejas Freddy (2013)

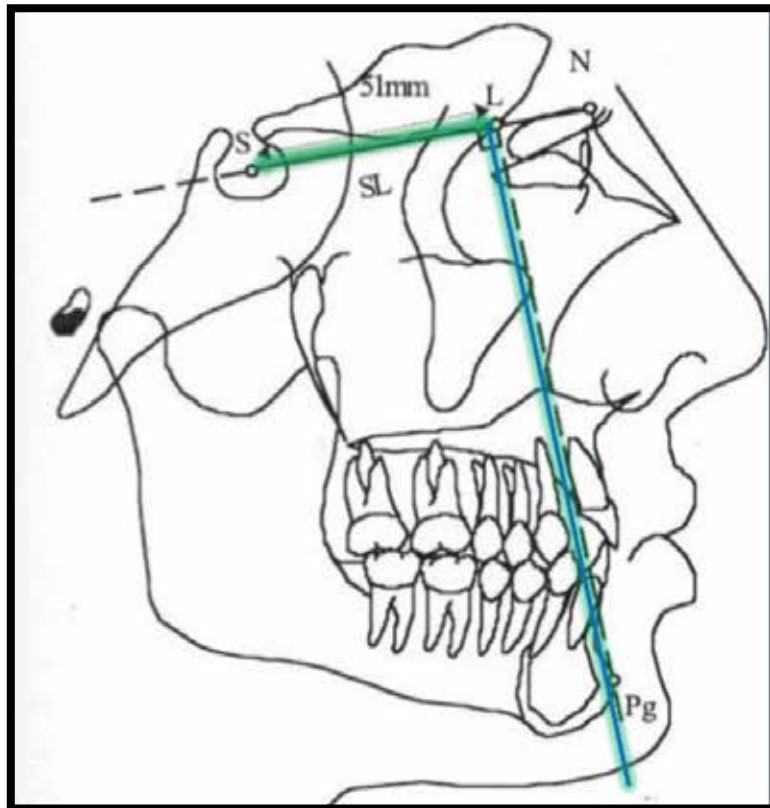
Segmento SL: Es la distancia entre el punto Silla y el punto L. Ver **Figura 7**.

Norma: 51mm.

Interpretación:

- El punto L se obtiene trazando una línea perpendicular al plano S – N que intersecte con el pogonion (Pg).
- Indica la relación anteroposterior del Pg. Respecto a la base del cráneo. Varía en casos de hiperplasia o hipoplasia mandibular.
- En pacientes con rotación mandibular en sentido horario esta dimensión disminuye y con pacientes con rotación mandibular en sentido anti horario la dimensión aumenta.

Figura 7. Segmento SL



Fuente: Parejas Freddy (2013)

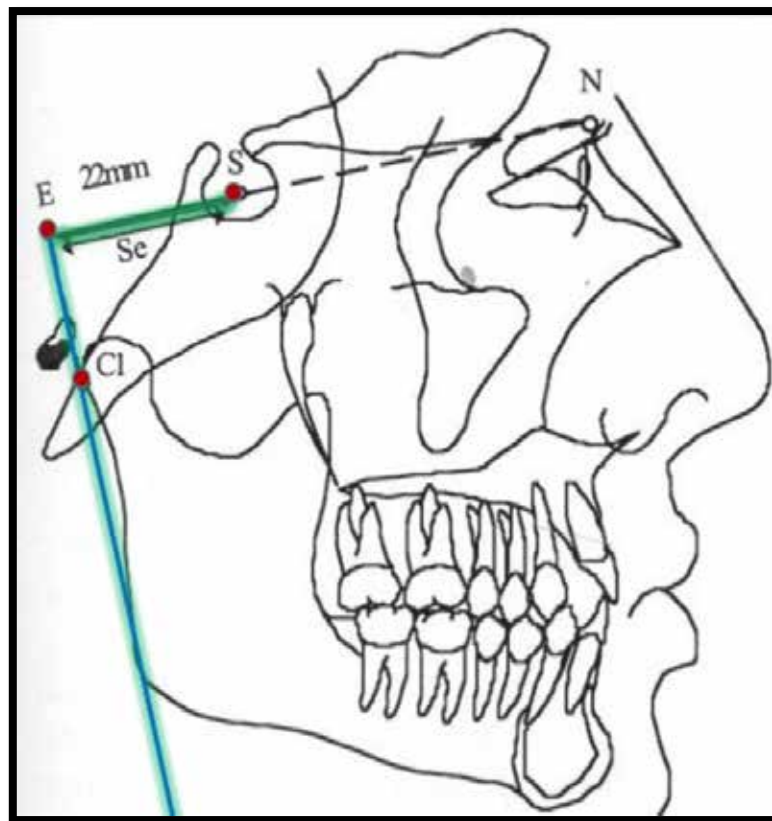
Segmento SE: Es la distancia entre el punto Silla y el punto E. Ver **Figura 8.**

Norma: 22mm.

Interpretación:

- El punto E se obtiene trazando una línea perpendicular al plano SN que pase por el contorno del cóndilo mandibular. (CI).
- Indica la ubicación anteroposterior del cóndilo de la mandíbula (en máxima intercuspidación) con respecto a la base del cráneo.
- Un valor aumentado indica que el borde posterior del cóndilo mandibular se encuentra ubicado más atrás con relación a la base craneal. Lo contrario indica una posición más adelantada.
- En pacientes braquiocefálicos la medida puede disminuir y en pacientes dolicocefálicos ésta medida puede ser aumentada.
- La posición adelantada o retrasada del cóndilo puede contribuir a la manifestación de una clase II o clase III esquelética.

Figura 8. Segmento SE



Fuente: Parejas Freddy (2013)

· **Relación vertical mandibulocraneal.**

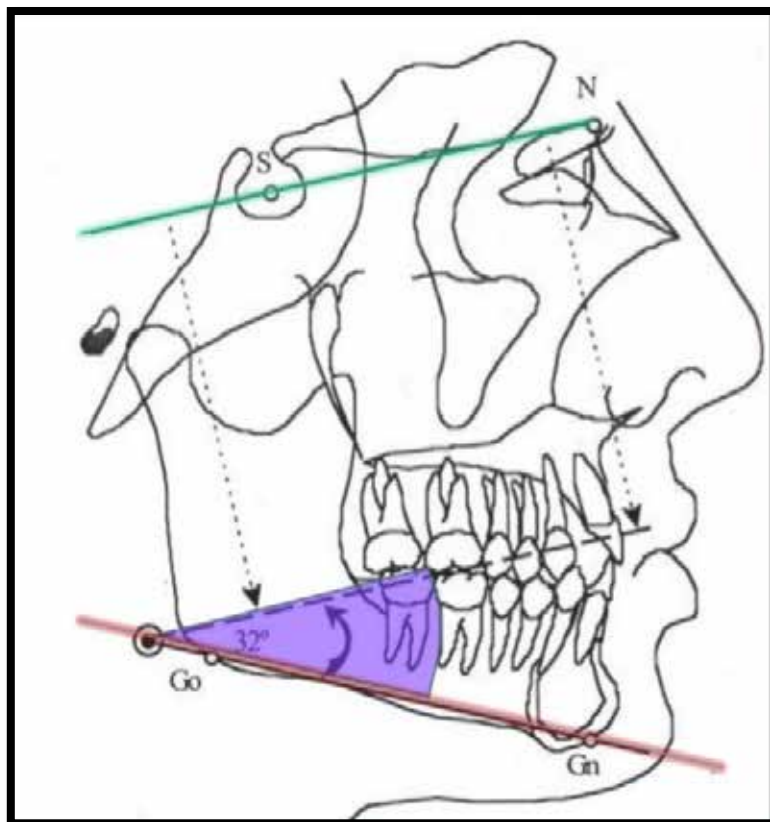
Ángulo Go – Gn S – N. Es el ángulo formado por los planos (Go – Gn) y el plano SN (Silla - Nasion). Ver **Figura 9**.

Norma: 32°.

Interpretación:

- Indica la dirección de crecimiento.
- Un ángulo mayor a 32° refleja un crecimiento vertical, común en los pacientes leptoprosopicos, pacientes con mordida abierta anterior esquelética.
- Un ángulo menor a 32° indica crecimiento horizontal, puede manifestarse en pacientes con mordida profunda.

Figura 9. Ángulo Go-Gn S-N



Fuente: Parejas Freddy (2013)

· **Relación oclusocraneal**

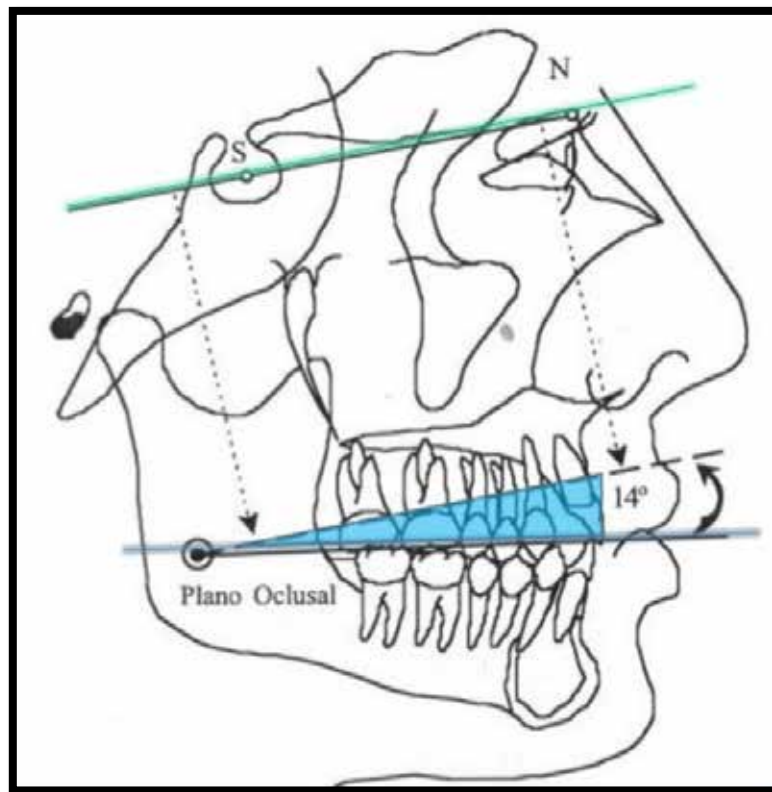
Plano oclusal a S – N: Es el ángulo formado por el plano oclusal y plano SN (Silla – Nasion). Ver **Figura 10**.

Norma: 14°.

Interpretación:

- Indica la inclinación del plano oclusal con respecto a la base del cráneo.
- Aumenta el ángulo en paciente con patrones de crecimiento vertical.
- Disminuye el ángulo en pacientes con patrón de crecimiento horizontal.

Figura 10. Plano oclusal a S-N



Fuente: Parejas Freddy (2013)

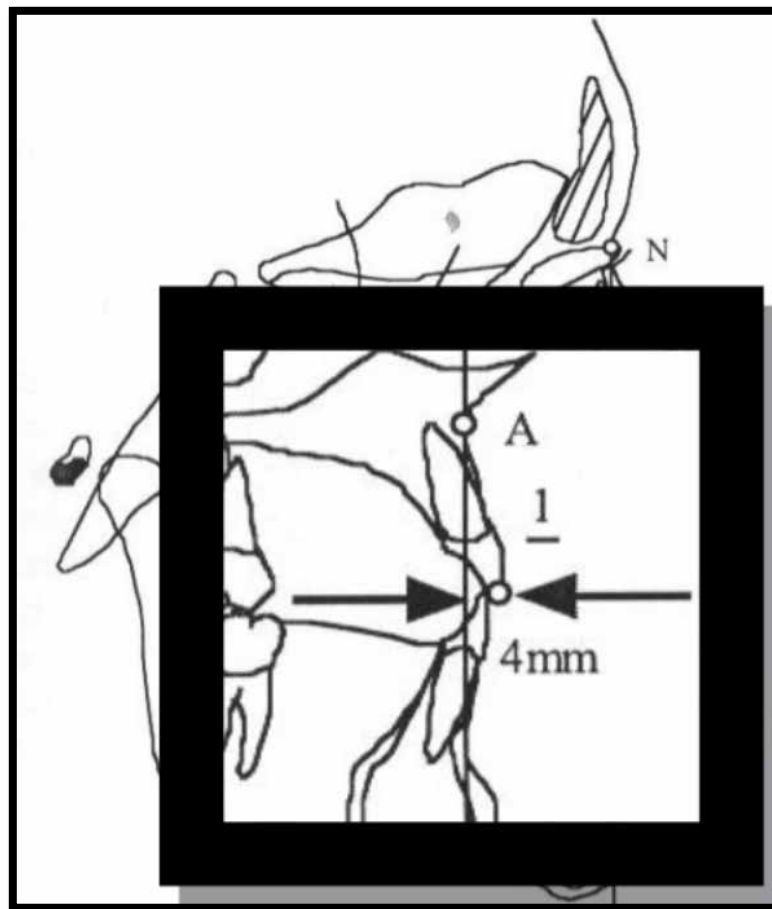
Ángulo 1 – NA: Es el ángulo formado por el eje longitudinal del incisivo superior y el plano NA (Nasion – Punto A). Ver **Figura 11**.

Norma: 22°.

Interpretación:

- Los valores aumentados indican protrusión del incisivo. (La protrusión se refiere a una medida anteroposterior, que proporciona la ubicación de la corona, sin conocer cómo se encuentra la inclinación axial del diente)
- Los valores disminuidos indican una retrusión del incisivo.

Figura 12. Distancia 1-NA



Fuente: Parejas Freddy (2013)

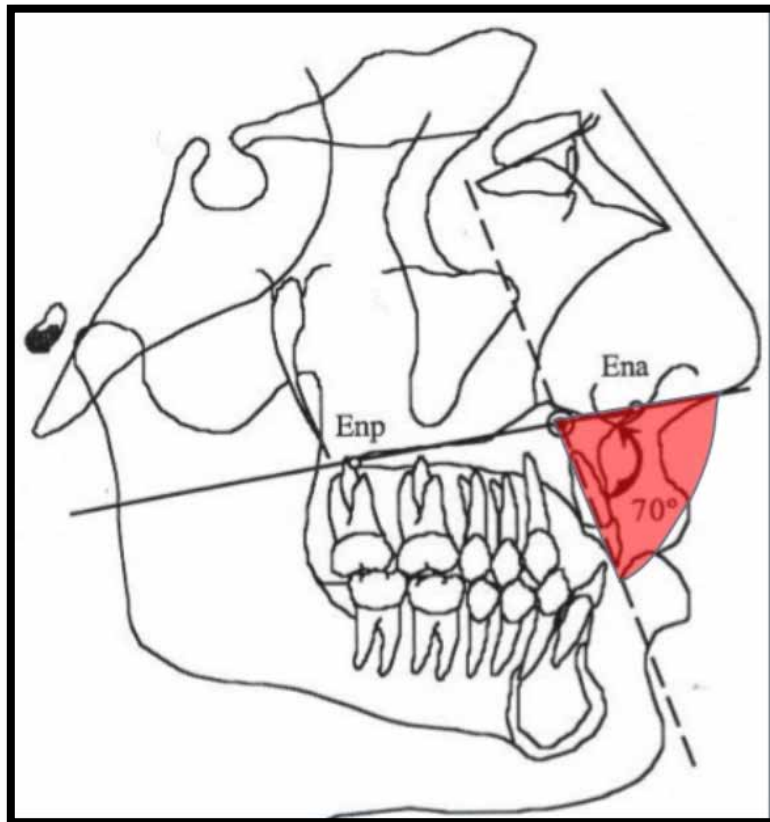
Ángulo Incisivo Superior – Plano Platino (1/Ena – Enp): Es el ángulo formado por el eje longitudinal del incisivo superior y el plano palatino (1/Ena – Enp). Ver **Figura 13.**

Norma: 70°.

Interpretación:

- Indica la inclinación anteroposterior del incisivo superior con respecto a su base ósea.
- Un ángulo mayor a 70° indica pro inclinación de los incisivos.
- Un ángulo menor a 70° indica retro inclinación.
- Es útil para determina la angulación de los incisivos cuando existen problemas esqueléticos severos.
- En los casos en los que el paciente vaya a ser sometido a un procedimiento ortodóntico – quirúrgico, se recomienda utilizar esta medida para eliminar la compensación natural de los incisivos.

Figura 13. Ángulo Incisivo Superior - Plano Platino



Fuente: Parejas Freddy (2013)

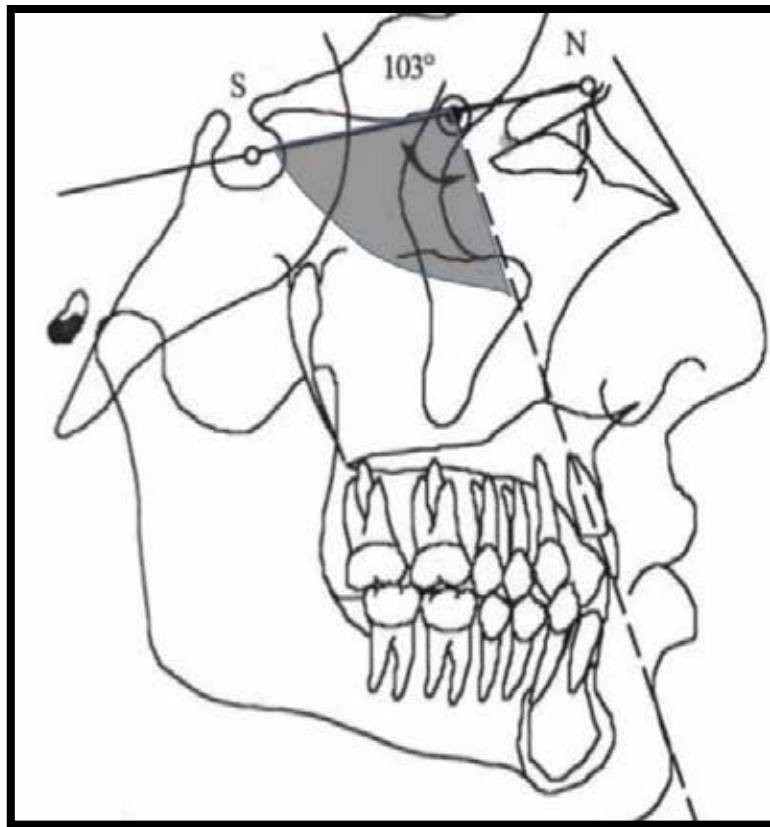
Angulación del incisivo superior con SN: Es el ángulo formado entre el eje axial del incisivo superior (U1) y el plano SN. Ver **Figura 14**.

Norma: 103°.

Interpretación:

- Establece el grado de inclinación de los incisivos superiores con respecto a la base del cráneo (SN).
- Un ángulo mayor a 103° indica pro inclinación de los incisivos.
- Un ángulo menor a 103° indica retro inclinación.
- Esta medida puede alterarse con la inclinación de la base del cráneo por lo que se sugiere corroborarla con el ángulo del incisivo superior con el plano palatino.

Figura 14. Angulación del incisivo superior con SN



Fuente: Parejas Freddy (2013)

· **Relación Inciso mandibular**

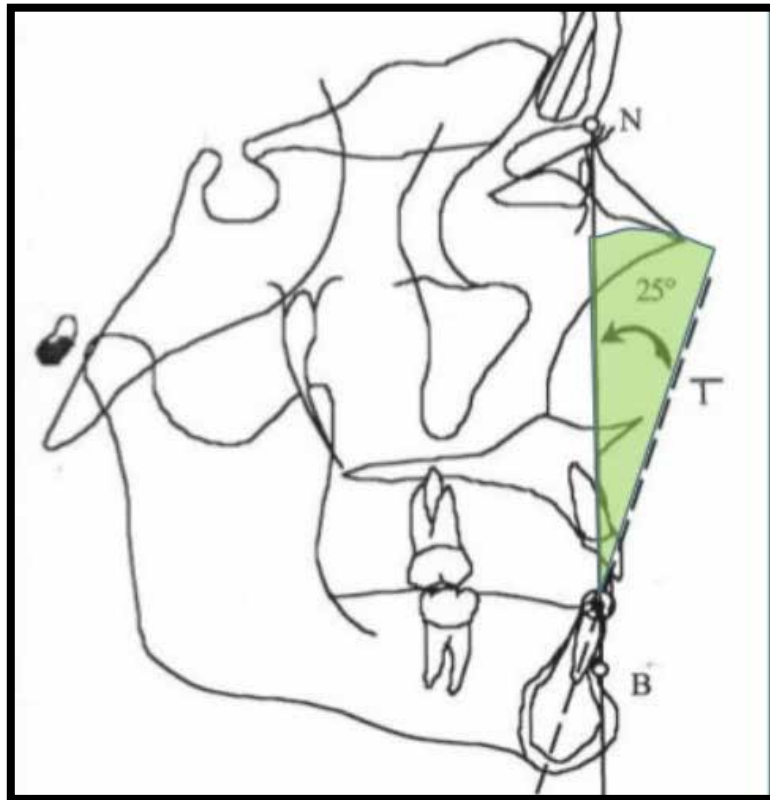
Ángulo 1 – NB: Es el ángulo formado por el eje longitudinal del incisivo inferior y el plano N-B. Ver **Figura 15**.

Norma: 25°.

Interpretación:

- Indica la inclinación anteroposterior del incisivo inferior.
- Un ángulo mayor a 25° indica pro inclinación del incisivo inferior.
- Un ángulo menor a 25° indica retro inclinación.
- Hay que considerar que una base craneal corta (Nasion ubicado hacia atrás) aumenta el ángulo
- Base craneal larga el ángulo disminuye.

Figura 15. Ángulo 1 - NB



Fuente: Parejas Freddy (2013)

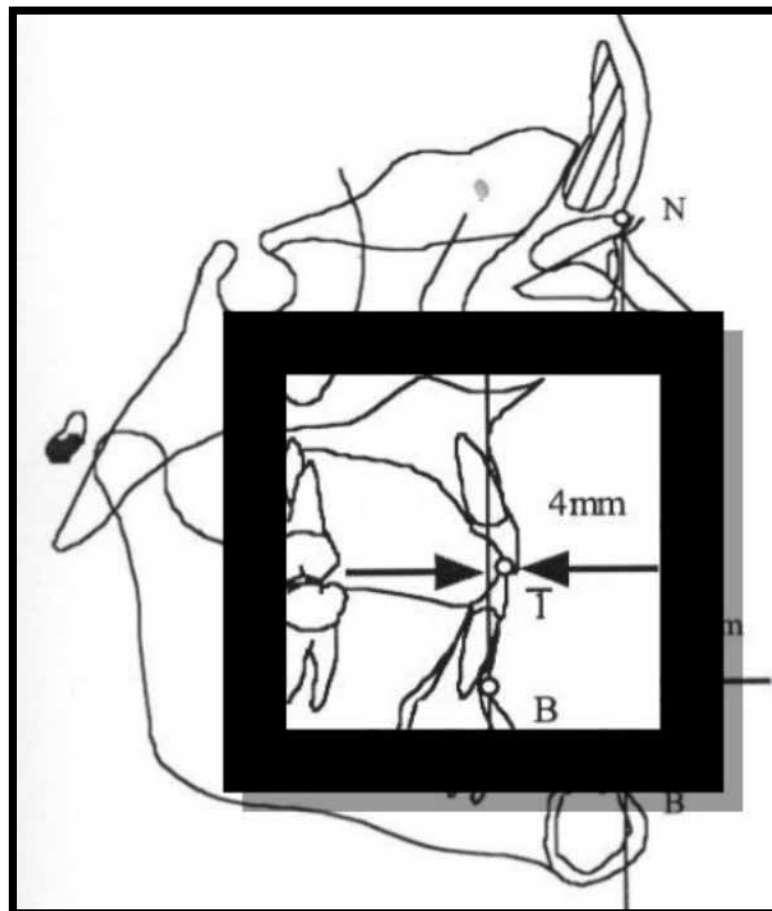
Distancia 1 – NB: Es la distancia entre el borde incisivo inferior y el plano NB.
Ver **Figura 16.**

Norma: 4mm.

Interpretación:

- Indica la ubicación anteroposterior del borde incisal del incisivo inferior.
- Una distancia mayor a 4mm indica protrusión del incisivo inferior.
- Una distancia menor a 4mm indican retrusión del incisivo.

Figura 16. Distancia 1-NB



Fuente: Parejas Freddy (2013)

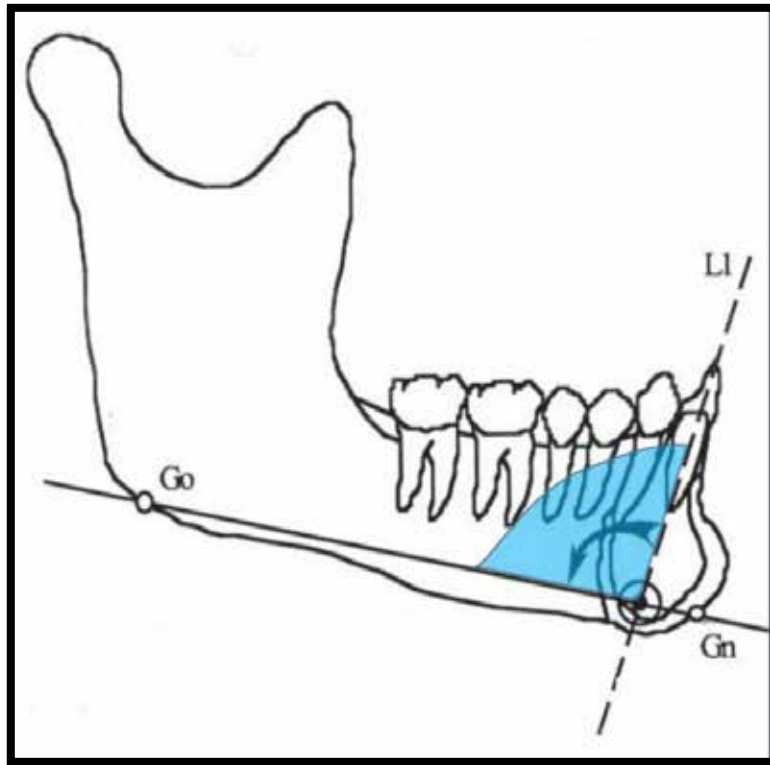
Incisivo Inferior al plano Go – Gn (L1 – Go – Gn): Es el ángulo formado por el eje longitudinal del incisivo inferior y el plano mandibular. Ver **Figura 17**.

Norma: 90°.

Interpretación:

- Indica la inclinación del incisivo inferior con respecto a su base ósea.
- Un ángulo mayor a 90° indica pro inclinación del incisivo inferior.
- Un ángulo menor a 90° indica retro inclinación.

Figura 17. Incisivo Inferior al plano Go-Gn



Fuente: Parejas Freddy (2013)

· **Relación interincisal**

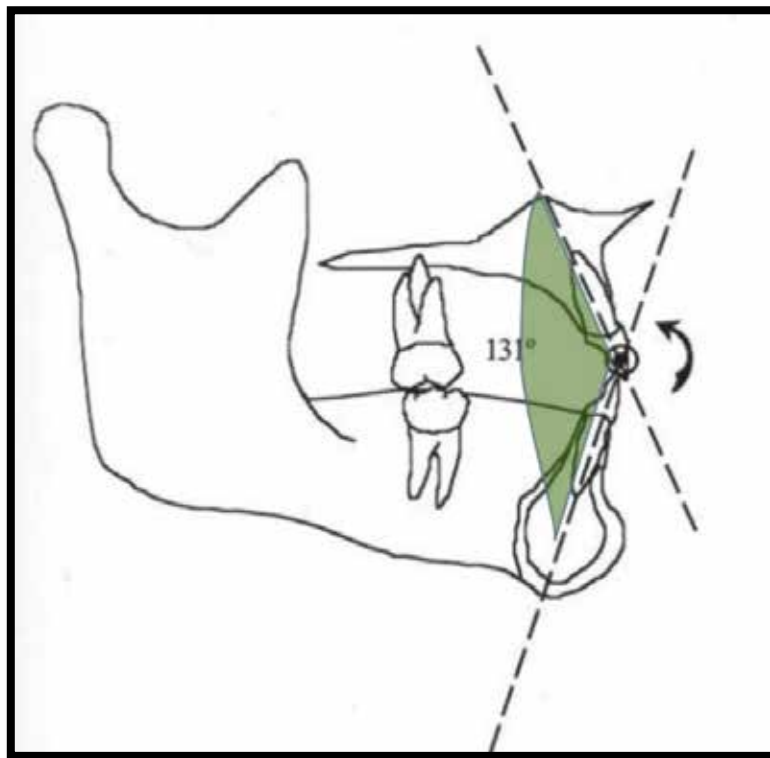
Ángulo Interincisal: Es el ángulo formado por los ejes longitudinales de los incisivos superior e inferior. Ver **Figura 18**.

Norma: 131°.

Interpretación:

- Indica la relación angular del eje longitudinal del incisivo superior en relación al eje longitudinal del incisivo inferior.
- Un ángulo mayor a 131° indica retro inclinación de incisivos.
- Un ángulo menor a 131° indica pro inclinación.
- Esta medida no es precisa si el ángulo está aumentado por causas de uno de ellos, ya sea incisivos superiores o inferiores.
- Un ángulo cerrado en pacientes biprotrusivos.
- Un ángulo abierto en pacientes con perfiles planos.

Figura 18. Ángulo interincisal



Fuente: Parejas Freddy (2013)

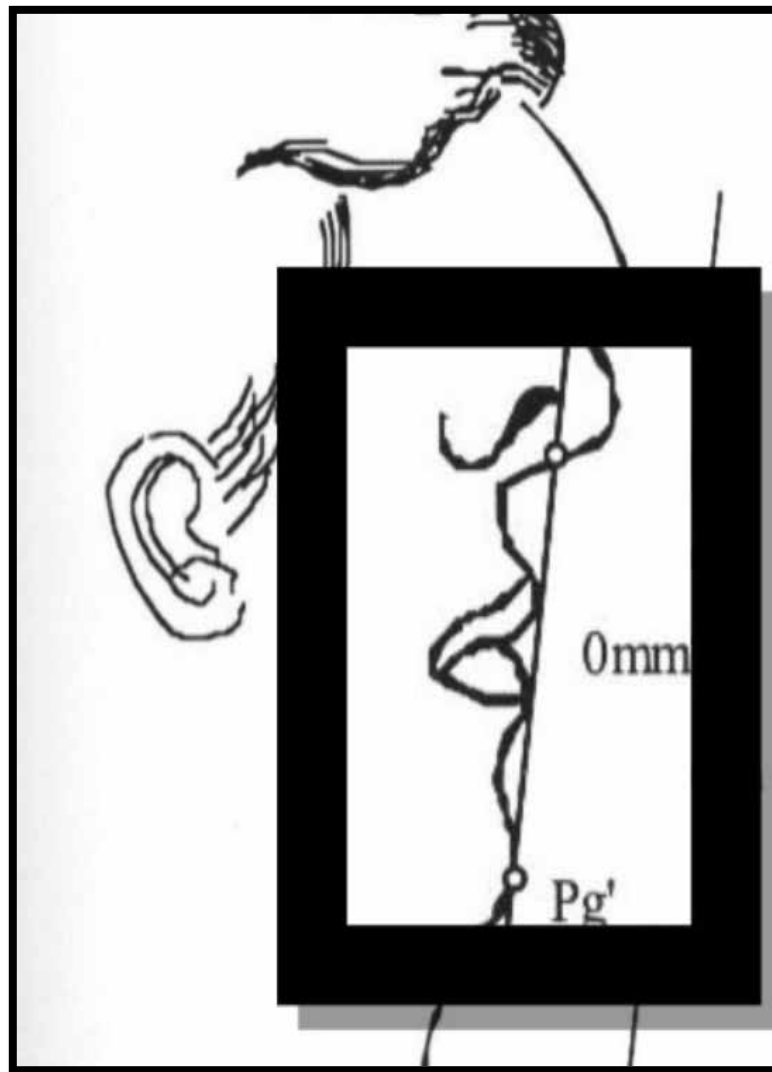
Línea S: Es la línea que une el Pogonion blando (Pg') con un punto ubicado en donde termina la S de la nariz. Ver **Figura 19**.

Norma: 0mm.

Interpretación:

- En un paciente que presenta un perfil armónico, labios superior e inferior se encuentran en contacto con la línea S.
- Cuando los labios se encuentran por delante (valores positivos) indica protrusión labial.
- Si los labios se encuentran por detrás (valores negativos) indica unos labios retrusivos.

Figura 19. Línea S



Fuente: Parejas Freddy (2013)

Cefalometría de Ricketts

Robert Murray Ricketts presentó un análisis cefalométrico en 1960. Procuró desarrollar un sistema de magnitudes que definiese en valores numéricos las tendencias del crecimiento facial, las proporciones dentarias, la posición del mentón y el maxilar y finalmente la estética facial. Con el tiempo nuevas medidas fueron incorporadas al análisis sumando un total de 33 factores. Esos factores o medidas cefalométricas fueron agrupados en seis (6) grupos o campos:

- Campo 1: Relación dentaria.
- Campo 2: Relación maxilomandibular.
- Campo 3: Relación dentoesquelética.
- Campo 4: Relación estética.
- Campo 5: Relación cráneo facial.
- Campo 6: Estructuras internas.

Para el desarrollo de este trabajo de grado, se tomaron en cuenta los factores más representativos agrupándolos en 5 grupos, estos son:

- Grupo 1: Clase esquelética.
- Grupo 2: Maxilar inferior.
- Grupo 3: Maxilar superior.
- Grupo 4: Dientes.
- Grupo 5: Estética

A continuación, se explica con más detalle cada análisis que se usó dentro de los diferentes grupos.

- **1er Grupo: Clase esquelética**

Convexidad Facial: Distancia entre punto A y plano facial (Nasion – Pogonio).

Ver **Figura 20**.

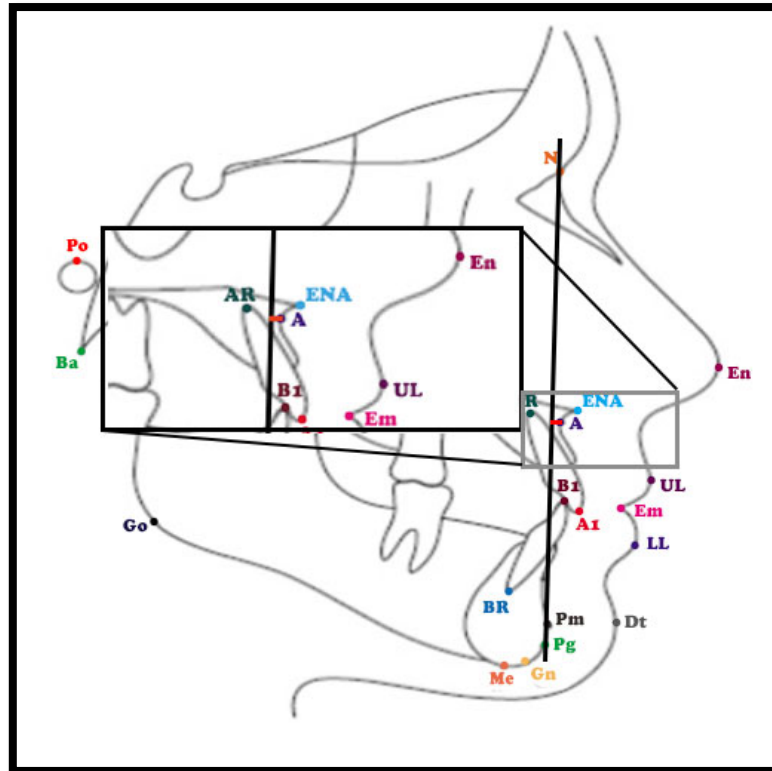
Norma: $2\text{mm} \pm 2\text{mm} + 0.2\text{mm}$ por año.

Interpretación:

- Medidas mayores a 2mm pertenecen a Clase II.

- Medidas menores a 2mm pertenecen a Clase III.

Figura 20. Convexidad Facial



Fuente: Rodríguez Sinaí (2017)

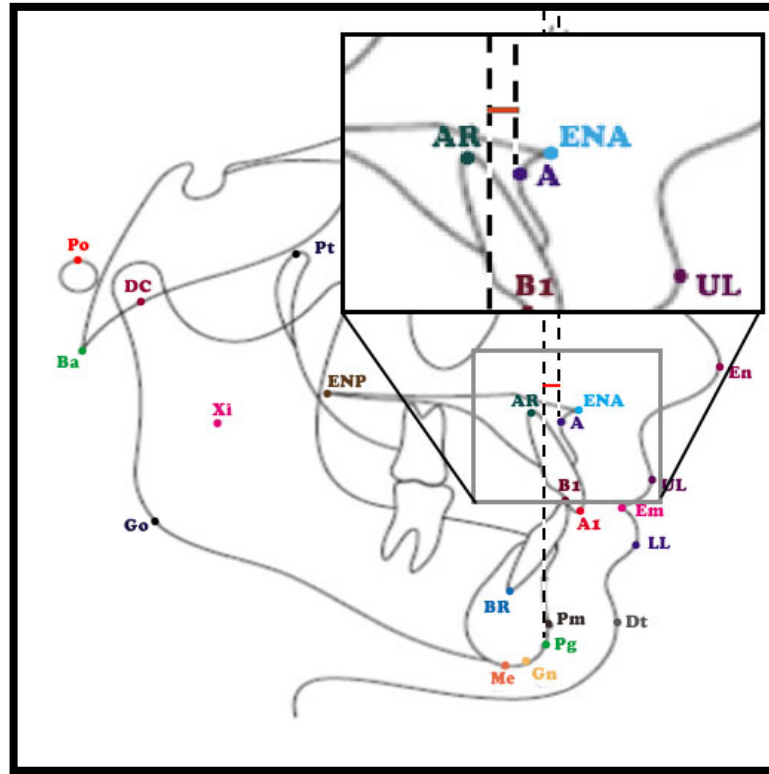
Distancia A – Pg (Wits): Se usa para saber la clase esquelética del paciente. Ver **Figura 21.**

Norma: $4\text{mm} \pm 2\text{mm}$.

Interpretación:

- Primero se debe trazar una recta que pase por el punto A.
- Segundo se debe trazar una recta que pase por el punto Pogonio, que a su vez sea paralela a la recta trazada anteriormente la cual atravesaba el punto A.
- Una vez trazadas las 2 rectas se procede a medir la distancia que existe entre ella de forma perpendicular para así obtener la distancia.

Figura 21. Distancia A-Pg (Wits)



Fuente: Rodríguez Sinaí (2017)

· **2do Grupo: Maxilar Inferior**

Eje Facial: formado por la unión del Eje Facial con Ba-N (Basion – Nasion). Ver **Figura 22.**

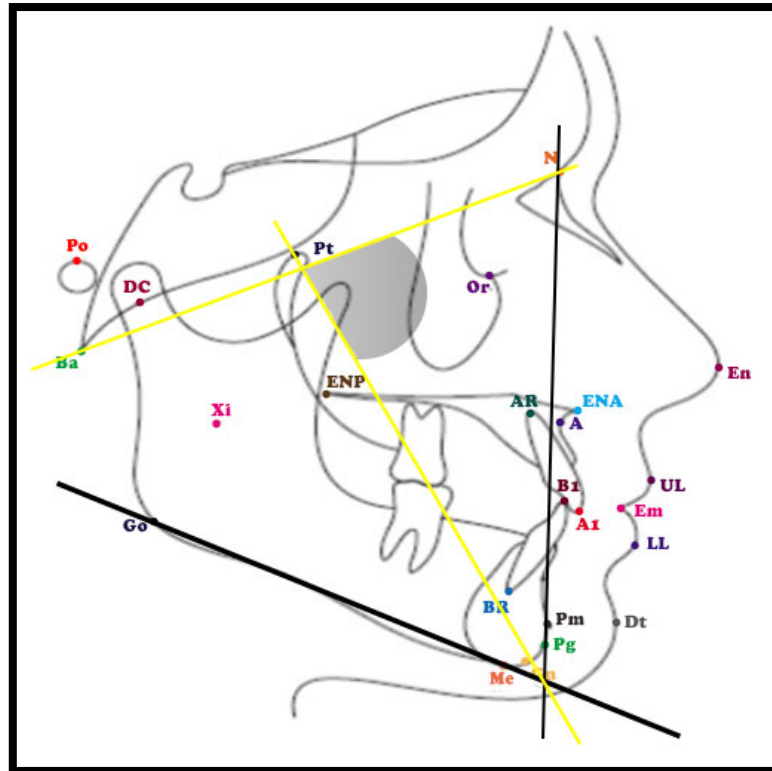
Norma: $90^\circ \pm 3^\circ$.

Interpretación:

- Para conseguir el eje facial primero se debe trazar las rectas N-Pg (Nasion- Pogonio) y Go-Me (Gonio – Mentoniano), donde se intersecten estas rectas es desde donde se trazará la recta que pasara por el puto Pterigoideo (Pt).
- El ángulo se formará con la unión del eje facial con el plano N-Ba
- Todo lo que gire en contra de las agujas del reloj es positivo.

- Todo lo que gire en sentido de las agujas del reloj es negativo.

Figura 22. Eje Facial



Fuente: Rodríguez Sinaí (2017)

Profundidad Facial: Ángulo que se forma con el Plano de Frankfort (Porion - Orbitario) y Plano facial (Nasion -Pogonio). Ver **Figura 23**.

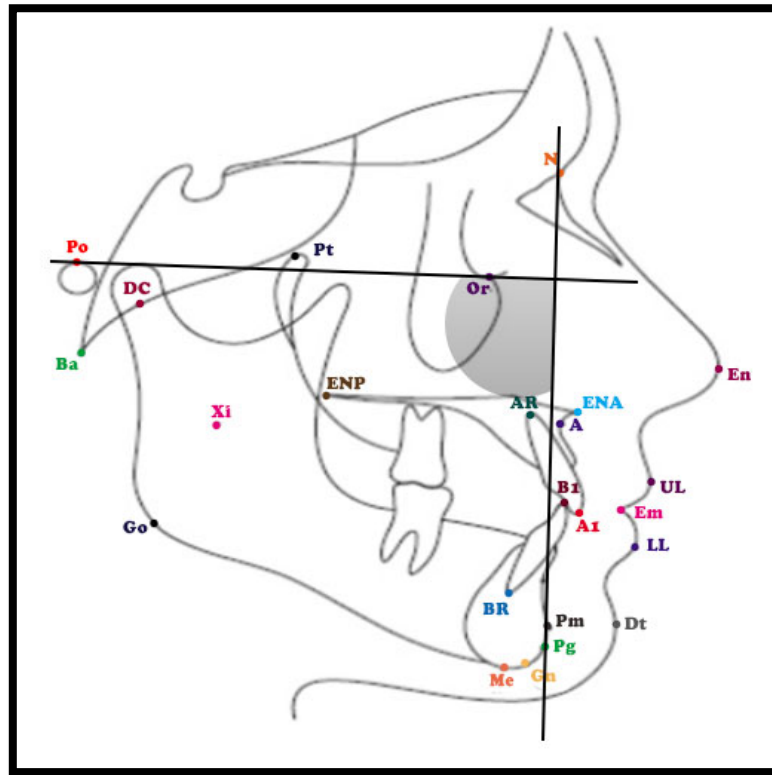
Norma: $87^\circ \pm 3$ a los 9 años, $+1^\circ$ cada 3 años.

Interpretación:

- Primero se debe conseguir el plano de Frankfort que está formado por el punto Porion y el punto Orbitario.
- Luego se debe conseguir el plano facial, que es la recta formada por los puntos Nasion y Pogonio
- Ayuda a ver quién tiene la culpa del crecimiento esquelético.

- Valores mayores indican una mandibular avanzada, mientras que valores menores indican una mandíbula deficiente.

Figura 23. Profundidad Facial



Fuente: Rodríguez Sinaí (2017)

Ángulo del Plano Mandibular: Ángulo que se forma entre el Plano de Frankfort (Ba –Or) y el Plano mandibular (Go – Me). Ver **Figura 24**.

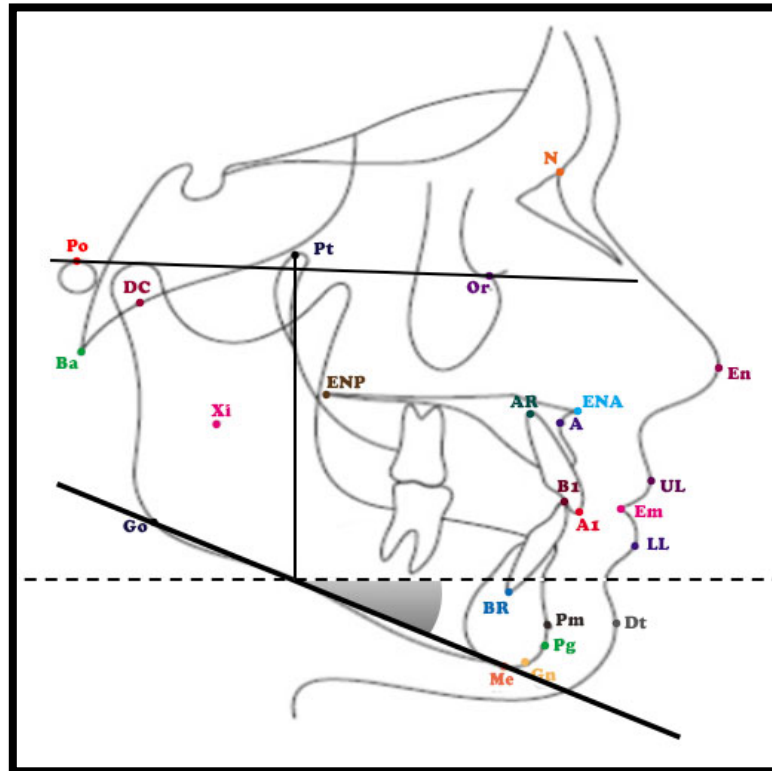
Norma: $26^\circ \pm 4^\circ$ a los 9 años - 1° cada 3 años.

Interpretación:

- Indica la dirección del crecimiento.
- Un ángulo aumentado refleja un crecimiento vertical. En pacientes con mordida abierta anterior esquelética, es frecuente encontrar esta medida aumentada

- Contrariamente, un ángulo cerrado indica un crecimiento horizontal, y puede estar presente en pacientes con una mordida profunda.
- Siempre se mide la mandíbula, no la proyección de Frankfort

Figura 24. Ángulo del Plano Mandibular



Fuente: Rodríguez Sinaí (2017)

Altura Facial Inferior: Ángulo formado por Xi y Espina Nasal Anterior (Xi – ENA) y Xi - Pm. Ver **Figura 25**.

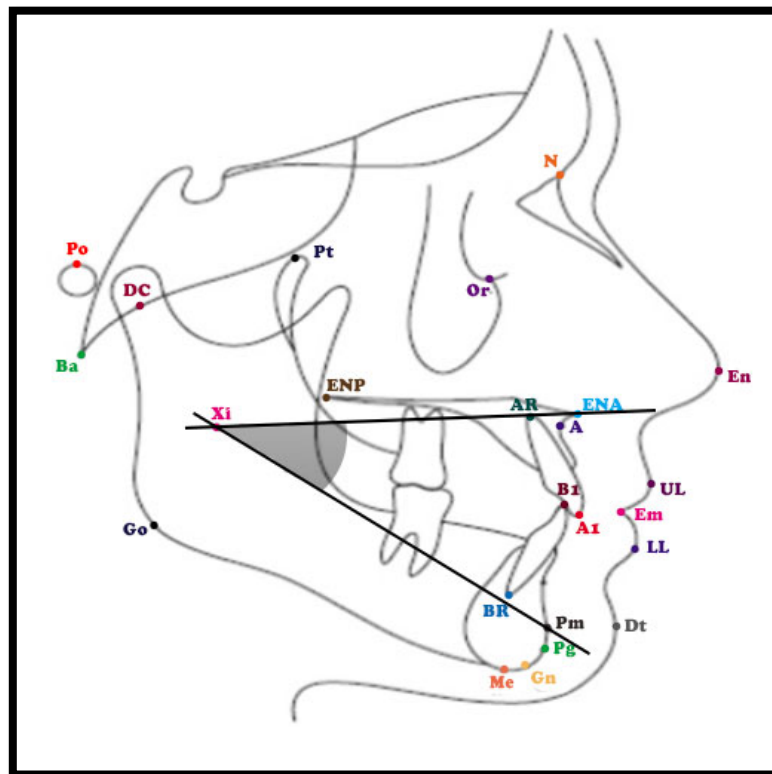
Norma: $47^\circ \pm 4^\circ$.

Interpretación:

- Para poder calcular Xi primero se deben contar con los valores para R1, R2, R3 y R4.
- Variaciones en su valor deber ser atribuidas al tratamiento.

- Todas aquellas maniobras que abren el eje facial hacen lo mismo con la altura inferior.
- Un ángulo mayor que la norma indica divergencia entre la mandíbula y el maxilar. Corresponde a un patrón dólico facial.

Figura 25. Altura Facial Inferior



Fuente: Rodríguez Sinaí (2017)

Arco Mandibular: Formado por la intersección del Eje Condilar (DC) - Xi con la Prolongación distal del eje del cuerpo de la mandíbula (Xi - Pm). Ver **Figura 26**.

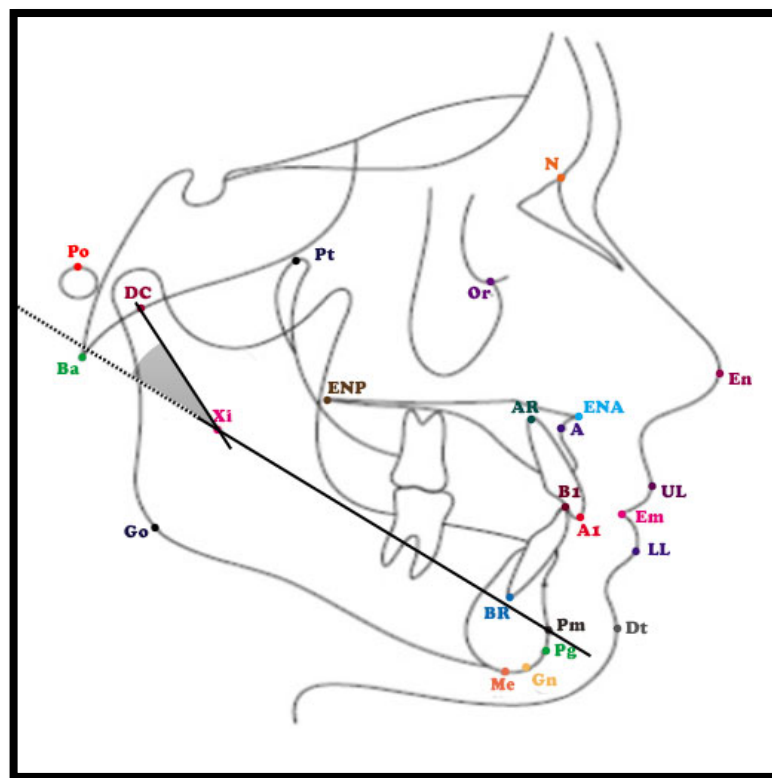
Norma: $26^\circ \pm 4$ a los 8.5 años, $+ 0.5^\circ$ por año.

Interpretación:

- Indica el grado de inclinación del cóndilo y el desarrollo mandibular.

- Un valor alto indica mandíbula de patrón prognático y tendencia a la supraoclusión esquelética (Braquicefálico - mordida cerrada mandibular).
- Un patrón bajo indica un valor vertical retrognático con tendencia a mordida abierta esquelética de origen mandibular (Dolicefálico mordida abierta mandibular).

Figura 26. Arco Mandibular



Fuente: Rodríguez Sinaí (2017)

· **3er Grupo: Maxilar Superior**

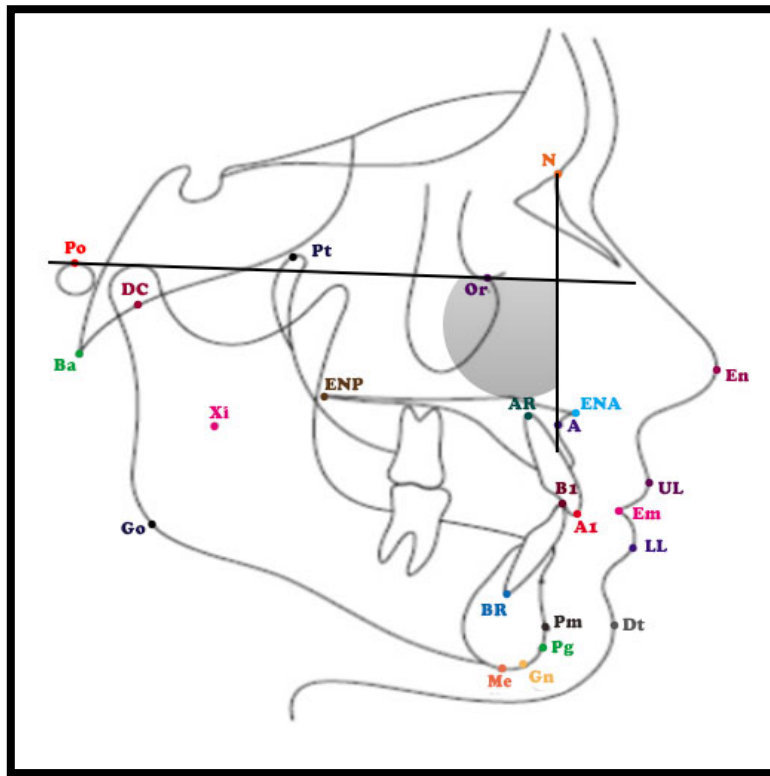
Profundidad Maxilar: Ángulo entre Plano de Frankfort (Porion - Orbitario) y Nasion – Punto A (N –A). Ver **Figura 27**.

Norma: $90^\circ \pm 3^\circ$.

Interpretación:

- Indica localización del maxilar en sentido anterosuperior.
- Mide de quien es la culpa a nivel esquelético.
- Valores superiores indican una protrusión esquelética del maxilar superior.
- Un ángulo menor de 90° nos habla de una retrusión esquelética del maxilar superior.

Figura 27. Profundidad Maxilar



Fuente: Rodríguez Sinaí (2017)

Altura maxilar: Ángulo formado entre Nasion - Centro Facial y el Punto A (N-CF-A). Ver **Figura 28**.

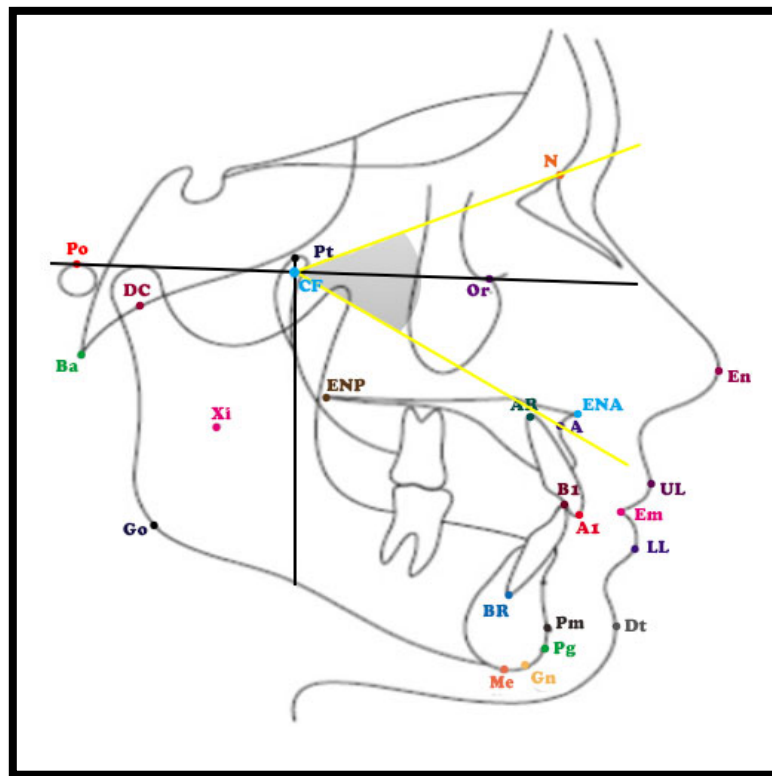
Norma: $53^\circ \pm 3^\circ$, + 0.4° por año.

Interpretación:

- Indica la posición del maxilar superior en el plano vertical.

- Sirve para diagnosticar mordidas abiertas esqueletal.
- Un valor cerrado nos indica un maxilar corto, probablemente responsable de una mordida abierta maxilar, una altura del tercio medio corta, caras cortas, colapsos maxilares.
- Un valor aumentado es indicativo de tercios medios aumentados, mordidas profundas maxilares, sonrisas gingivales por excesos verticales del maxilar, caras largas.

Figura 28. Altura Maxilar



Fuente: Rodríguez Sinaí (2017)

· **4to Grupo: Dientes**

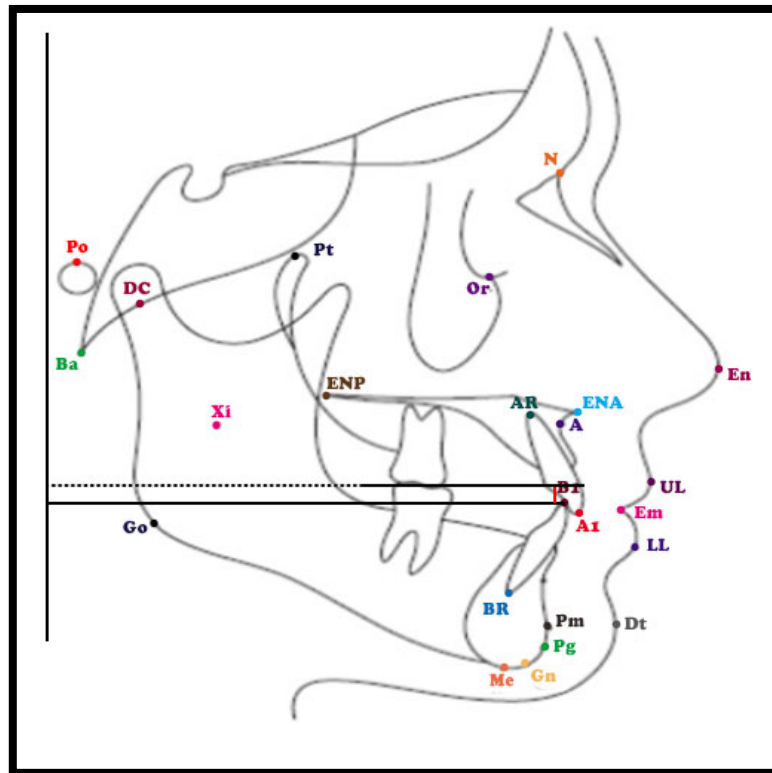
Incisivo Inferior 1 A –Pg (Plano dentario): Distancia entre el borde del Incisivo Inferior y plano A-Pg. Ver **Figura 29**.

Norma: 1mm ± 2mm.

Interpretación:

- Dictamina si el overbite se debe al incisivo superior o inferior.
- Indica la relación y proporción vertical de la maloclusión valorando la cantidad de sobremordida vertical.
- A mayor overbite, mayor distracción condilar durante las excursiones funcionales.
- A menor overbite, menor funcionalidad y gran porcentaje de daño articular.
- Los valores mayores se consideran mordidas profundas.
- Los valores menores de 0mm se consideran mordidas abiertas.

Figura 30. Incisivo Inferior al Plano de Oclusión (1- Plano Oclusal)



Fuente: Rodríguez Sinaí (2017)

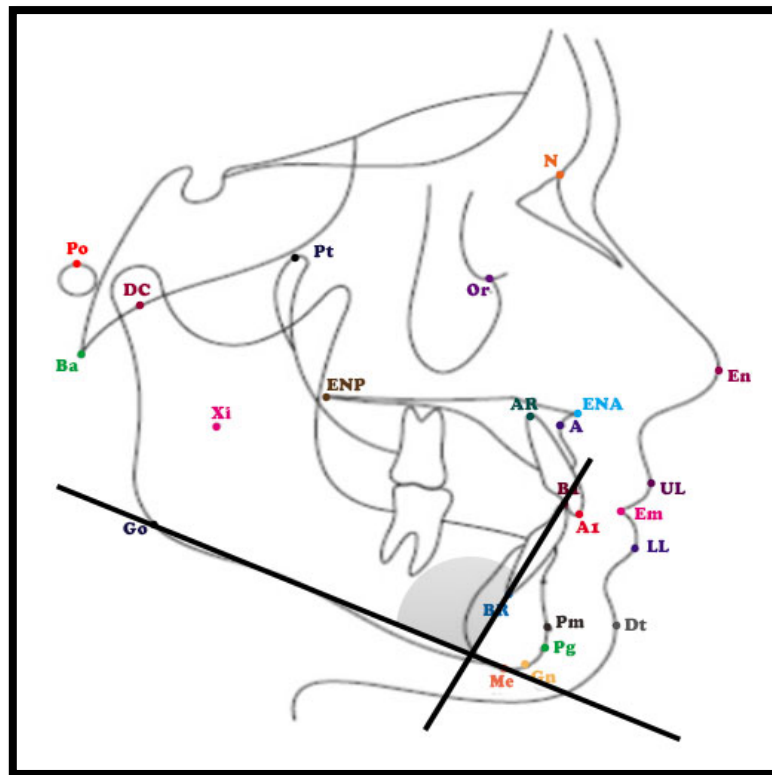
1 Plano Mandibular (IMPA): Ángulo formado por el eje del incisivo inferior con el plano mandibular (Go-Me). Ver **Figura 31**.

Norma: $90^\circ \pm 5^\circ$.

Interpretación:

- Por cada mm que se retrocede en el 1 A-Pg se retroceden 2° en el IMPA.
- Indica la estabilidad del incisivo inferior dentro de la sínfisis mentoniana.

Figura 31. 1 Plano Mandibular (IMPA)



Fuente: Rodríguez Sinaí (2017)

1 Plano Ba-N: Ángulo formado por el plano N-Ba (Nasion –Basion) y el eje del incisivo superior. Ver **Figura 32**.

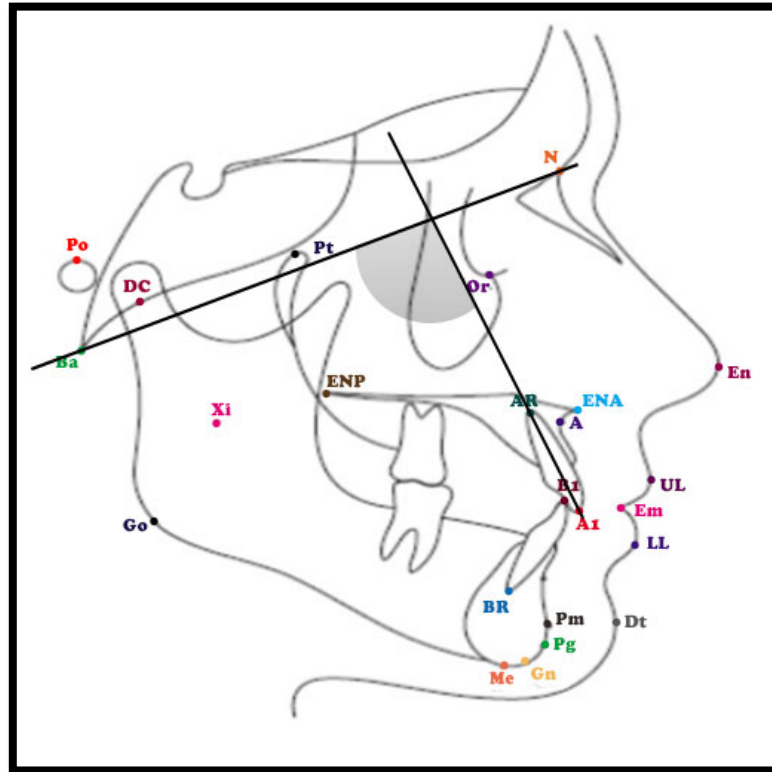
Norma: Es el valor del eje facial – 5° .

Interpretación:

- Indica la estabilidad del incisivo superior en el maxilar.
- Si el ángulo se encuentra aumentado es sinónimo de protrusión del incisivo superior.

- Si el ángulo se encuentra disminuido da señales de retruccion del incisivo superior.

Figura 32. 1 Plano Ba-N



Fuente: Rodríguez Sinaí (2017)

· **5to Grupo: Estética**

Exposición del Incisivo Superior: Distancia desde el incisivo superior a la comisura labial superior. Ver Figura 33.

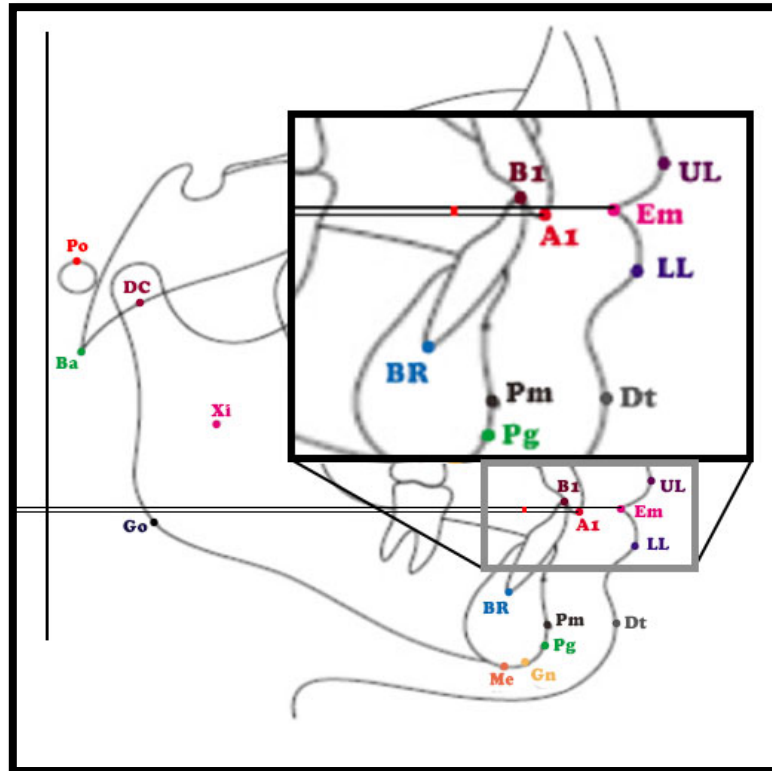
Norma: 2.5mm ± 3mm.

Interpretación:

- Para saber la exposición del incisivo superior.
- Se requiere la vertical verdadera para trazar una perpendicular a altura de la comisura labial superior.
- Al igual que el paso anterior se traza una perpendicular, pero a altura del incisivo superior por su punto A1.

- Luego se mide la distancia entre la recta formada y el punto A1.

Figura 33. Exposición del Incisivo Superior



Fuente: Rodríguez Sinaí (2017)

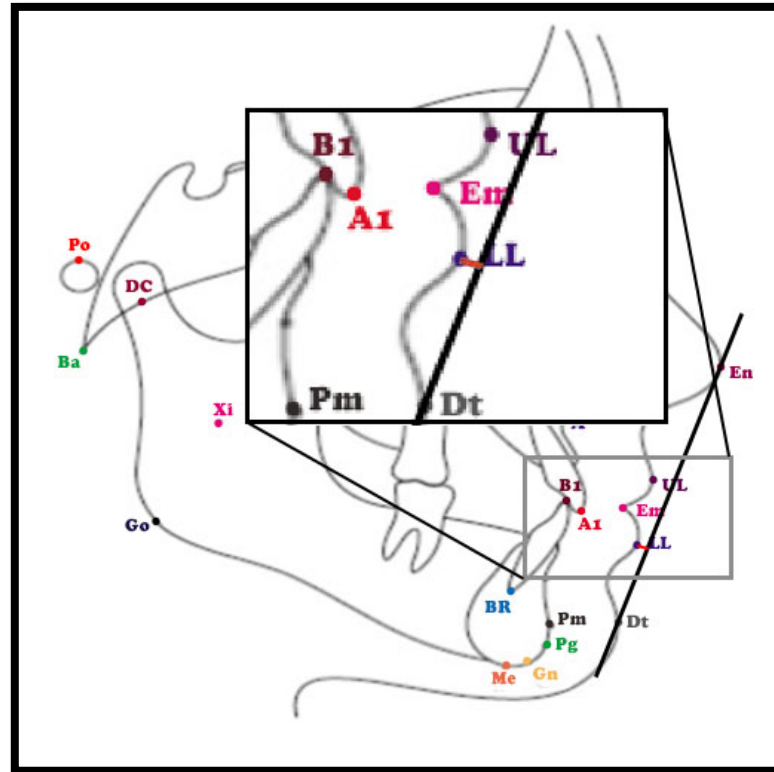
Labio inferior con el plano E: Distancia desde la parte más anterior del labio inferior al plano estético(En-Dt). Ver **Figura 43**.

Norma: $-2\text{mm} \pm 2\text{mm}$ a los 8.5 años, -0.2mm por año.

Interpretación:

- Indica relación entre nariz, labio, mentón.
- Lo normal es que el labio inferior este por detrás del plano E.
- Si el labio se encuentra por delante del plano E presentaría proquelia
- Y el incisivo estaría protruido.
- A nivel estético daría imagen de un perfil convexo aun cuando esqueléticamente no lo sea.

Figura 34. Labio inferior con el plano E



Fuente: Rodríguez Sinaí (2017)

Ángulo nasolabial inferior: Formado entre el perfil inferior de la nariz y el perfil del labio superior. Ver **Figura 35**.

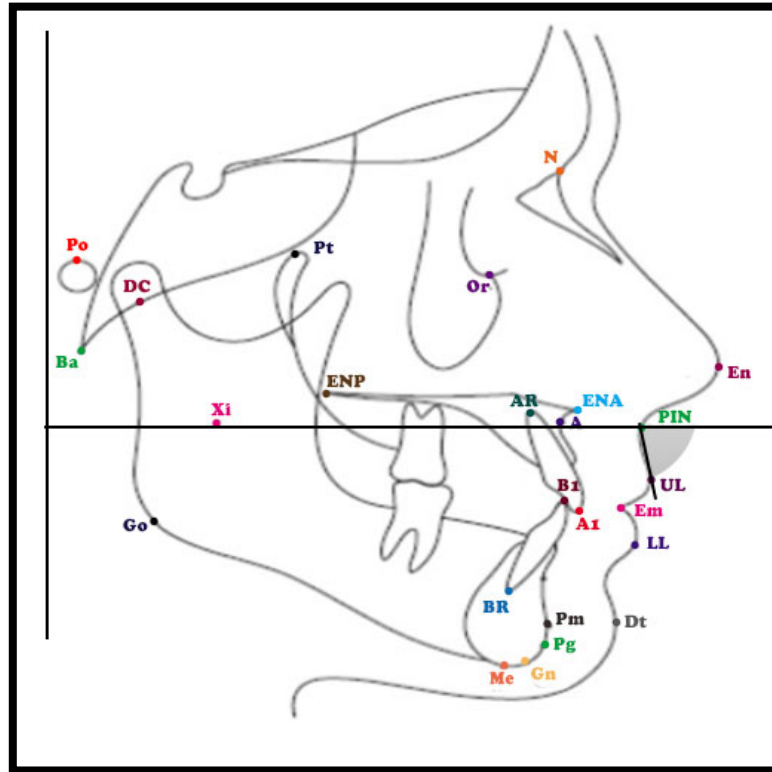
Norma: $85^\circ \pm 5^\circ$.

Interpretación:

- Indicador muy importante de la posición del incisivo superior.
- Se necesita la vertical verdadera para trazar una perpendicular a nivel del punto PIN
- Luego se traza una línea que pase por el perfil anterior del labio superior.
- Si el ángulo es más cerrado da señal de protrusión del incisivo superior.
- Si el ángulo es menos cerrado indica retrucción del incisivo superior.
- A nivel estético si el ángulo se encuentra disminuido daría imagen de un perfil convexo aun cuando esqueléticamente no lo sea.

- Estéticamente si el ángulo se encuentra aumentado daría imagen de un perfil cóncavo aun cuando esqueléticamente no lo sea.

Figura 35. Ángulo nasolabial inferior



Fuente: Rodríguez Sinaí (2017)

2.4. Definición de Términos Básicos

JSON (JavaScript Object Notation – Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Está constituido por dos estructuras:

- Una colección de pares de nombre/valor, en varios lenguajes esto es conocido como objetos, registro, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores, en la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una u otra forma.

DOM: El modelo de Objetos del Documento (DOM) es una interfaz de programación de aplicaciones (API) para documentos HTML y XML. Define la estructura lógica de los documentos y el modo en que se accede y manipula un documento. En la especificación del DOM, el término “documento” se utiliza en un sentido amplio. XML se utiliza cada vez más como un medio para representar muchas clases diferentes de información que puede ser almacenada en sistemas diversos y mucha de esta información se vería, en términos tradicionales, mas como datos que como documentos. Sin embargo, XML presenta estos datos como documentos, y se puede usar el DOM para manipular estos datos.

Framework: Es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

POSTMAN: Es una herramienta para hacer peticiones a las APIs con la que se pueden generar colecciones de peticiones que permiten probar las aplicaciones de una manera rápida y sencilla, siendo las colecciones carpetas donde se almacenan las peticiones, ayudándonos estructurar ya sea por recursos, módulos o como se desee.

Puntos cefalométricos:

- **Basion (Ba):** Punto posteroinferior del hueso occipital en el margen anterior del foramen magno.
- **Eje Condilar (Dc):** Punto situado en el centro del cuello del cóndilo, sobre el plano Ba-N.
- **Espina Nasal Anterior (ENA):** Punto localizado en la extremidad anterior y superior del maxilar.
- **Espinal Nasal Posterior (ENP):** Punto más posterior del maxilar.
- **Gnation (Gn):** Punto formado por la intersección de los planos mandibular de Margolis y el facial de Downs.
- **Gonion (Go):** Localizado por la intersección de los planos mandibular de Margolis y Ramal, el centro del contorno posteroinferior de la mandíbula.
- **Mentoniano (Me):** Punto medio más inferior, situado sobre la curvatura inferior de la sínfisis mentoniana.

- **Nasion (N):** Localizado en la unión de la sutura frontonasal con los huesos propios de la nariz.
- **Orbitario (Or):** Punto más inferior del contorno de la órbita. El punto orbitario se localiza en la unión del reborde orbitario externo con el piso de la órbita.
- **Punto A ó subespinal:** Punto más posterior de la concavidad anterior en el perfil óseo del maxilar, ubicado entre la espina nasal anterior y el reborde alveolar.
- **Punto B ó supramental:** Punto más posterior de la concavidad anterior en el perfil óseo de la mandíbula, ubicado entre pogonion y el reborde alveolar.
- **Punto D:** Se localiza en el centro de la sínfisis mentoniana.
- **Punto E:** Localizado en el plano SN, trazando una perpendicular a este que pase por la parte más posterior del cóndilo mandibular.
- **Punto incisivo inferior (1):** Es el punto más vestibular de la corona del incisivo inferior más anterior.
- **Punto L (L):** Localizado en el plano SN (silla-nasion) trazando una perpendicular de este que pase por el punto Pg.
- **A1:** Borde incisal del incisivo central superior.
- **AR:** Porción apical del incisivo central superior.
- **B1:** Borde incisal del incisivo central inferior.
- **BR:** Porción apical del incisivo central inferior.
- **Pogonion (Pg):** Es el punto más anterior de la curvatura anterior de la sínfisis mentoniana.
- **Porion (Po):** Punto más superoanterior del conducto auditivo externo.
- **Protuberancia Mentoneana (Pm):** Punto localizado en la curvatura del borde anterior de la sínfisis, cuando la curvatura pasa de cóncava hacia convexa.
- **Pterigoideo (Pt):** Punto localizado en la intersección de las paredes posterior y superior de la fisura pterigomaxilar.

- **Silla (S):** Este punto se encuentra localizado en el centro de la silla turca del esfenoides.
- **Xi:** Punto situado en el centro de la rama ascendente de la mandíbula, localizado en el foramen mandibular donde el nervio mandibular penetra en la mandíbula.
- **R1:** Parte más profunda del borde anterior de la rama mandibular.
- **R2:** Proyección de R1 en el borde posterior de la rama mandibular.
- **R3:** Porción más profunda de la escotadura sigmoidea.
- **R4:** Proyección de R3 en borde inferior de la rama de la mandíbula.
- **Centro del Cráneo (CC):** Intersección del eje facial con Ba – N.
- **Centro Facial (CF):** Intersección del plano Po-Or con la vertical Pterigoidea
- **Punto En:** Punto más anterior de la nariz.
- **Punto UL:** Punto más anterior del labio superior.
- **Punto LL:** Punto más anterior del labio inferior.
- **Punto Em:** Unión de comisura labial superior e inferior.
- **Punto Dt:** Porción más anterior del mentón blando.

CAPÍTULO III

MARCO METODOLÓGICO

3.1. Tipo de Investigación

La presente investigación es de tipo proyecto especial, ya que la Universidad Pedagógica Experimental Libertador (UPEL) la señala como “el desarrollo de software, prototipos y productos tecnológicos en general”. Estos proyectos especiales se componen de actividades, uso de recursos materiales o financieros; requieren objetivos específicos, necesidades o problemas y se deben realizar en un tiempo determinado.

3.2. Diseño de la Investigación

Según Arias (1999), define el diseño de la investigación como “la estrategia que adopta el investigador para responder al problema planteado”. En este caso se utilizará un diseño de investigación de campo ya que Fideas G. Arias (2012) define:

La investigación de campo es aquella que consiste en la recolección de todos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin importar o controlar variables alguna, es decir, el investigador obtiene la información, pero no altera las condiciones existentes. De allí su carácter de investigación no experimental.

Claro está, en una investigación de campo también se emplea datos secundarios, sobre todo los provenientes de fuentes bibliográficas, a partir de los cuales se elabora el marco teórico. No obstante, son los datos primarios obtenidos a través del diseño de campo, los esenciales para el logro de los objetivos y la solución del problema planteado. (p.31).

3.3. Nivel de Investigación

Para el desarrollo del presente trabajo de grado se utilizará un nivel de investigación descriptivo, ya que se trabajará con datos verdaderos tomados directamente de la realidad, y evaluándolos para dar un resultado oportuno de lo que

se espera; según Sabino (1986) dice que “la investigación de tipo descriptiva trabaja sobre realidades de hechos, y su característica fundamental es la de presentar una interpretación correcta. Para la investigación descriptiva, su preocupación primordial radica en descubrir algunas características fundamentales de conjuntos homogéneos de fenómenos, utilizando criterios sistemáticos que permitan poner de manifiesto su estructura o comportamiento. De esta forma se pueden obtener las notas que caracterizan a la realidad estudiada”. (p.51).

3.4.Población y Muestra

La población es el conjunto de individuos de la misma clase que limita la investigación. Según Tamayo y Tamayo (1997), “La población se define como la totalidad del fenómeno a estudiar donde las unidades de población poseen una característica común la cual se estudia y da origen a los datos de la investigación”. (p114). Mientras que la muestra es la que genera los datos con los que se van a trabajar dentro de nuestro software, Tamayo y Tamayo afirman que la muestra “es el grupo de individuos que se toma de la población, para estudiar un fenómeno estadístico”. (p.38).

En este caso se tomará como población a los docentes de odontología de la Universidad José Antonio Páez que imparten la asignatura de ortodoncia y ortopedia dentofacial. Mientras que la muestra será un 1 docente, el cual será el encargado de probar al API para el análisis de radiografías y así verificar si cumple con todos los estándares de la Universidad José Antonio Páez. El tipo de muestra es un muestreo intencional u opinático el cual Arias F. (2012) define como “selección de los elementos con base en criterios o juicios del investigador”. (p.85)

3.5.Técnicas e instrumentos de recolección de información

Para Bizquera (1990) las técnicas de investigación son “aquellos medios técnicos que se utilizan para registrar observaciones y facilitar el tratamiento de las mismas”. (p.28). En este caso se utilizará como técnica de recolección de información la entrevista mixta y la observación directa, ya que a través de la entrevista se puede obtener la información necesaria para el levantamiento de las bases de lo que se desea

para el sistema, según Arias (2006) las entrevistas son “más que un simple interrogatorio, es una técnica basada en un diálogo o conversación "cara a cara", entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida " (p.73). Mientras que con la observación se podrán obtener resultados sin involucrarse con la muestra escogida de la población; Arias (2006) define el método de observación “como aquella que se realiza cuando el investigador observa de manera neutral sin involucrarse en el medio o realidad en la que se realiza el estudio”. (p. 69).

3.6.Fases Metodológicas

Existen numerosas propuestas de metodología para desarrollar software. Tradicionalmente estas metodologías se centran en el control del proceso, estableciendo rigurosamente las actividades, herramientas y notaciones al respecto, dado estas reglas estas metodologías se caracterizan por ser rígidas y dirigidas por la documentación que se genera en cada una de las actividades desarrolladas.

Metodología XP

La programación extrema o eXtreme Programming (XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck; la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Características de la Metodología XP:

- Pone más énfasis en la adaptabilidad que en la previsibilidad.
- Se aplica de manera dinámica durante el ciclo de vida del software.
- Es capaz de adaptarse a los cambios de requisitos.
- Los individuos e interacciones son más importantes que los procesos y herramientas.

Fases de la Metodología XP:

Las fases de la metodología XP está representada por una serie de pasos los cuales se especifican a continuación.

1. Fase I - Planificación del proyecto:

- 1.1. Historias de usuarios: Definir las historias de usuarios con el cliente, estas tienen la misma finalidad que los casos de uso, pero con algunas diferencias. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen, también se usan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario
- 1.2. Release Planning: Es el plan de publicaciones, es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa.
- 1.3. Iteraciones: Cada iteración tiene una duración de 3 semanas aproximadamente y al principio de cada una se debe decidir que reléase planning será implementada.
- 1.4. La velocidad del proyecto: Es una medida que representa la rapidez con la que se desarrollará el proyecto, para ella es necesario evaluar la cantidad de historias de usuarios que se pueden aplicar por cada iteración.
- 1.5. Programación en parejas: Es recomendable ya que incrementa la productividad y la calidad del software, permitiendo dividir el trabajo, uno codificando mientras el otro analizando.
- 1.6. Reuniones diarias: Será donde los desarrolladores expongan sus problemas, soluciones e ideas de formas conjunta.

2. Fase II – Diseño

- 2.1. Diseños simples: Se procura hacer la API sin complicaciones posibles para conseguir un diseño de fácil entendimiento e implementación, que a la larga costara menos tiempo y esfuerzo desarrollar.
- 2.2. Glosario de términos: Estos ayudaran a comprender el diseño y facilitara sus posteriores ampliaciones y la reutilización del código
- 2.3. Riesgos: Se sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que suponen los problemas.

2.4. Funcionabilidad extra: No se deben añadir, aunque se piensen que en un futuro será utilizada

2.5. Refactorizar: Es el mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionabilidad.

3. Fase III – Codificación

Es donde se desarrollarán las historias de los usuarios de la mano del cliente, esta debe hacerse atendiendo a estándares de codificación ya creados, ya que así se mantiene un código consistente, facilitando su comprensión y escalabilidad.

4. Fase IV – Pruebas

Será donde se compruebe el funcionamiento de los softwares que vayamos implementando en nuestro software. Esto se revisará a través de las pruebas de caja blanca y caja negra a las cuales será sometido el software para comprobar su correcto funcionamiento y poder corregir posibles errores que surjan. Para ellos se utilizará la siguiente estructura de pruebas

Tabla 2. Pruebas

Prueba	Resultado
Comparar modelos de Sistemas con SQL y NoSQL	
Prueba del Web Service	
Prueba de la API-REST	
Simulación de solicitud de análisis de Stainer con la API-REST	
Simulación de solicitud de análisis de Stainer con la API-REST	

Fuente: González Luis (2017)

CAPÍTULO IV

RESULTADOS

En el siguiente capítulo se muestran los resultados obtenidos a partir de la metodología utilizada que fue programación extrema (XP). Se evidencia con detalle las historias de usuarios, requerimientos funcionales y no funcionales, diagrama de clases, diagramas de casos de uso, diccionario de datos, pruebas de caja negra y caja blanca y se expondrán los beneficios obtenidos.

4.1. Fase 1: Planificación del proyecto

4.1.1. Historias de usuarios

Las historias de usuarios representan la información necesaria para comenzar a desarrollar el Software de trazado, esta información se consiguió por medio de los métodos de recolección de información mencionados en el capítulo III, estos son primeramente la entrevista mixta y la observación directa al momento de las pruebas.

4.1.1.1. Aplicación de método de recolección de información

Por medio de una entrevista mixta la cual constaba de unas sencillas preguntas que se iban formulando con mejor fluidez a medida que se interactuaba con el Docente Oswaldo Mejías, profesor de ortodoncia y ortopedia dentofacial de la Universidad José Antonio Páez, se pudo obtener la información base para el proyecto la cual se necesitaba desarrollar en la universidad y así establecer una comparación de los softwares de trazado que existen en el mercado actual.

La entrevista se llevó a cabo dentro de la Universidad José Antonio Páez y tenía la siguiente estructura:

- **Pregunta 1:** ¿Qué es un software de trazado?

Respuesta: Es aquel software encargado de realizar trazados ortodónticos, análisis cefalométricos y procedimiento de imágenes diagnósticas visuales, así también se

usa para planes de tratamiento tanto con fines ortodónticos como de cirugía maxilofacial.

Esta pregunta se realizó con la finalidad de ver el conocimiento que se poseía sobre Software de trazado, para evitar confusiones que pudieran surgir por malas interpretaciones de alguna de las partes, al tener una idea del qué era lo que la Universidad solicitaba por Software de trazado, se tuvo una mejor idea acerca de lo que se debía realizar, y gracias a la explicación del profesor se pudo comprender el por qué la universidad requería un software de trazado dentro de sus instalaciones.

- **Pregunta 2:** ¿Conoce algún software de trazado en el mercado actual?

Respuesta: Si, Facad y Dolphin Imaging.

Con esta pregunta se pudo comprender si se tenía conocimiento de software de trazados en el mercado actual y de cómo funcionaban, teniendo como base estos softwares de trazado se pueden usar como referencia para el diseño de interfaces o para determinar funcionalidades que son las más comunes de encontrar en éstos.

- **Pregunta 3:** ¿Cuáles análisis cefalométricos debe realizar un software de trazado comercial actualmente?

Respuesta: Deben realizar muchos, depende de lo que se quiera estudiar, pero los más comunes son las cefalometrías de Steiner, la cefalometría de Ricketts, la cefalometría de McNamara y la cefalometría de Bjork- Jarabak

Con la siguiente pregunta se buscaba saber el alcance de los softwares de trazado comerciales actualmente, ¿cuáles eran las funcionalidades que poseían? y el ¿Por qué se usaban en la actualidad? Dando como resultado que se usaban por la cantidad de análisis cefalométricos que se realizaban a una velocidad bastante alta por medio de una interfaz agradable para el usuario.

- **Pregunta 4:** ¿Cuáles análisis cefalométricos debe realizar el software de trazado que se realizará a la Universidad José Antonio Páez?

Respuesta: Por ahora es bueno comenzar con las cefalometrías de Steiner y Ricketts.

Ya teniendo una idea del alcance de los softwares de trazado actuales, se comenzó a preguntar sobre ¿cuáles serían los métodos que se usarían dentro del software de trazado realizado dentro de la Universidad José Antonio Páez? Para así saber cuáles serían los análisis cefalométricos con los que se comenzaría a desarrollar el software y cuales se podrían añadir en el futuro cuando se actualice el sistema.

- **Pregunta 5:** ¿Conoce otras instituciones donde se pueda utilizar el software de trazado desarrollado para la Universidad José Antonio Páez?

Respuesta: Si, se puede llevar a la Universidad de Carabobo en donde también imparto clases.

Esta pregunta solo se usó para ver hasta donde se podría utilizar el software de trazado, y observar si no se planeaba limitar su uso solo para las instalaciones de la Universidad José Antonio Páez.

- **Pregunta 6:** ¿Esas instituciones usan los mismos análisis cefalométricos que se utilizan dentro de la Universidad José Antonio Páez?

Respuesta: Si

Al saber que otras instituciones cercanas emplean los mismos análisis cefalométricos que se utilizan dentro de la Universidad José Antonio Páez se plantea la posibilidad de hacer que el sistema pueda ser usado por otras instituciones y no solo a la cual está destinada.

- **Pregunta 7:** ¿Quiénes usarían el software de trazado dentro de la Universidad José Antonio Páez?

Respuesta: Estará destinado a los estudiantes de dicha facultad e institución bajo la supervisión de los profesores de ortodoncia.

Teniendo en consideración quienes serían los usuarios que usarían el software de trazado se tendría una idea del diseño que se debe usar de forma que sea amigable para ellos, ya sea teniendo poco o alto conocimiento acerca de los análisis cefalométricos y los pasos para realizarlos, a pesar de que los alumnos al usar el software estarán bajo supervisión de un profesor capacitado.

- **Pregunta 8:** ¿Quiénes considera usted que deberían utilizar el software de trazado al momento de realizar las pruebas?

Respuesta: Al momento de las pruebas es mejor utilizar directamente a profesores de ortodoncia que tienen un mayor conocimiento de cómo debería funcionar un software de trazado.

Esta pregunta solo se usó pensando en cómo realizar las pruebas al software de trazado, ayudándonos de la opinión del profesor y su experiencia en el uso de estos softwares conociendo el método más óptimo para realizar las pruebas del sistema, obteniendo una respuesta positiva diciendo que la mejor forma de probar el software sería con personal capacitado y con experiencia, en este caso, los profesores de ortodoncia de la Universidad José Antonio Páez.

4.1.2. Release Planning

Durante la realización del Release Planning, se establecieron las fechas de entrega ya pautadas para el desarrollo correcto del sistema (API-REST), por medio de las que se fueron probando las funcionalidades y óptimas condiciones de uso de dicha aplicación, teniendo en consideración las fechas límites de cada entrega, y dejando algunas para correcciones o para añadir nuevas funcionalidades que se pudieran dar con el paso del tiempo. Todo se pudo realizar gracia a la información obtenida mediante las historias de los usuarios ya establecidas previamente en la planificación y distribución de dichas fechas, lográndolo de manera eficiente. Estas fechas se pueden apreciar en el calendario entregas que se muestra a continuación con más detalle. Ver **Figura 36.**

codificar la estructura, funcionabilidad y confiabilidad de todo el sistema y así garantizar el éxito de éste en el transcurso del tiempo.

4.1.6. Reuniones diarias

Al estar trabajando una sola persona en el desarrollo de la API del Software de trazado no había la necesidad de reuniones diarias, más sin embargo si había reuniones semanales con los colaboradores del proyecto para ir evaluando el desarrollo de éste.

4.2. Fase II – Diseño

4.2.1. Diseños simples

Luego de aplicar la entrevista mixta para la recolección de datos y analizarlos se procedió a elaborar los diagramas adecuados para el correcto desarrollo de la API del software de trazado bajo el lenguaje unificado de modelado (UML) ya que este lenguaje nos ayudaba a visualizar, especificar, construir y documentar el sistema. Además de que basado en la información recolectada se pudieron establecer cuáles serían los requerimientos funcionales y no funcionales para que el sistema responda al problema planteado

Requerimientos funcionales:

Para el administrador del sistema:

- Poder crear nuevas cefalometrías de manera óptima usando las operaciones ya establecidas en el sistema, de tal forma que solo se necesiten definir su modelo y controlador para su uso.
- Crear los métodos HTTP (Hypertext Transfer Protocol) de manera rápida para su uso al momento de crear nuevas cefalometrías, gracias al sistema de rutas.
- Añadir nuevas operaciones matemáticas y analíticas para ser usadas posteriormente en las cefalometrías que se puedan ir incorporando.
- Establecer como se recibirán los datos desde la interfaz creada en la Fase I.

- Establecer como se devolverán los resultados a la interfaz creada en la Fase I.
- Añadir nuevas formas de evaluación a los resultados obtenidos a través de las funciones matemáticas.
- Realizar una interpretación de los resultados obtenidos con los valores estándares para cada cefalometría junto con su margen de error.

Para el usuario del sistema:

- Poder ingresar puntos bajo coordenadas (X; Y) para obtener resultados dependiendo los puntos ingresados.
- Poder ver una interpretación de los resultados obtenidos juntos con los valores estándares de cada cefalometría.
- La velocidad de respuesta debe ser lo más alta posible.
- El margen de error debe ser nulo.

Requerimientos no funcionales

- **Eficiencia:** La API debe ser capaz de procesar N transacciones por segundo. Toda funcionalidad del sistema y transacción debe responder al usuario en menos de 5 segundos. La API debe ser capaz de operar correctamente con grandes cantidades de solicitudes consecutivas. Los datos al ser modificados en la base de datos deben dar una respuesta inmediata.
- **Usabilidad:** El tiempo de aprendizaje del software debe ser el menor, ya que se estructuró de tal manera que no cause confusión. Además de que se crearon los manuales específicos para la API. En caso de producirse un error el sistema te avisa cual es el error y donde se produjo.
- **Dependibilidad:** La API cuenta con un tiempo de inicio bastante rápido y con un tiempo de reinicio inmediato al producirse algún cambio significativo para

la forma de analizar los datos. Además, la probabilidad de falla del sistema es mínima.

- Seguridad: Los datos solo pueden ser modificados o cambiados por los administradores, mientras que los usuarios solo pueden modificar los puntos.
- Compatibilidad: La API está diseñada para que se puedan conectar a ella indiferentemente del lenguaje en el que esté la interfaz que se conecta, ya que trabaja con JavaScript y devuelve datos en forma de JSON, que pueden ser leído por los lenguajes de programación.
- Validaciones: La API cuenta con sistema de validaciones para minimizar el margen de error a 0, trabajando en base solo a lo que pase las validaciones sin afectar el funcionamiento de la misma.

Se logró una buena planeación con respecto a la API para que así trabajara de la manera más rápida y optimizada posible, todo gracias a un buen diseño de base. Para el diseño se utilizó el Lenguaje de Unificado de Modelado (UML) ya que nos ayudaba a visualizar, especificar, construir y documentar un sistema de software

4.2.1.1. Diagrama de casos de uso

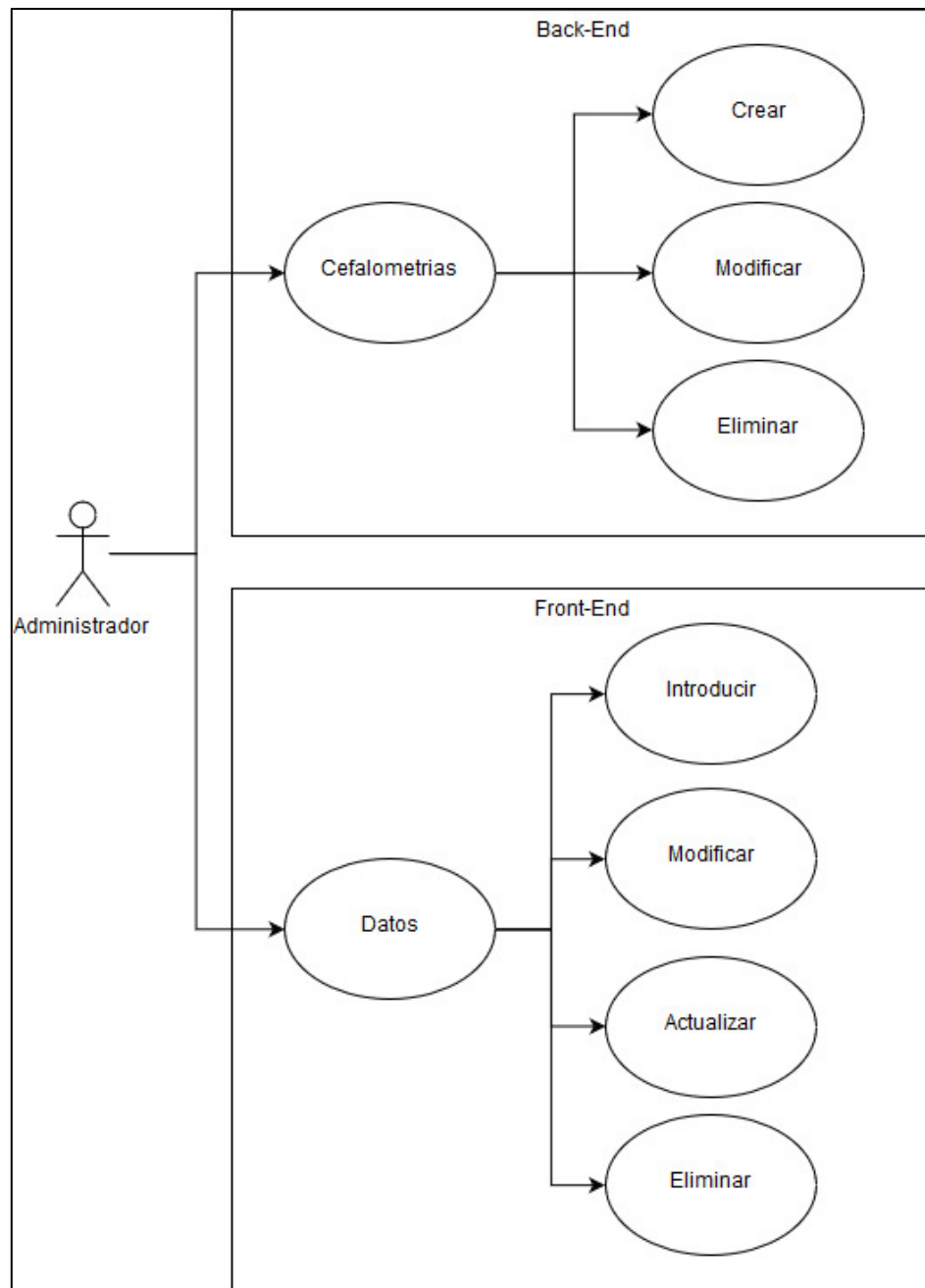
Para el desarrollo del sistema no se cuenta con el sistema de roles, pero se puede ver a que partes del sistema tiene acceso los usuarios y los administradores, tanto en el back-End (API) como en el front-end (Interfaz)

Casos de uso del Administrador

Aquí se puede observar como el administrador es aquella persona encargada del manejo del back-end de la aplicación, es decir, el que desarrolla la API; al ser su trabajo en la parte del servidor es él el que se encarga de agregar nuevos análisis cefalométricos a través de Node.js, a los cuales se les puede acceder desde la interfaz desarrollada en la Fase I. Su función también cumple con modificar análisis previos creados o eliminarlos si son necesarios. Para la parte de la interfaz el administrador puede acceder

desde la interfaz para ingresar datos para ser analizados, modificar resultados que ya se tengan o eliminar análisis previos que se hallan realizado.

Figura 37. Casos de uso del administrador

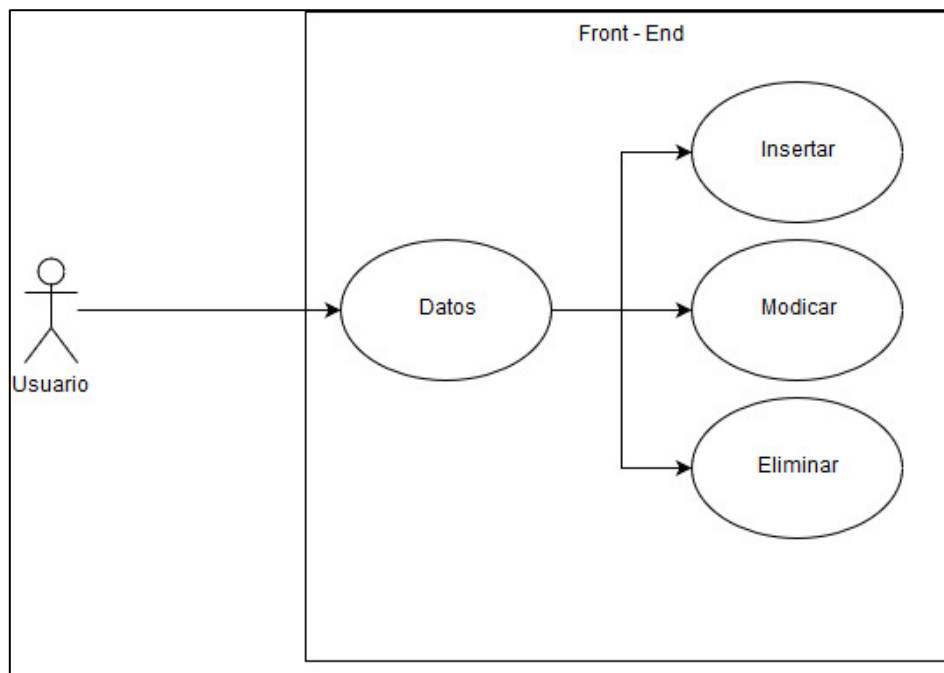


Fuente: González Luis (2017)

Casos de uso del Usuario

El usuario es aquel rol que se encargará sólo de insertar puntos para ser analizados dependiendo el análisis cefalométrico seleccionado, ésta persona será la encargada de introducir puntos, actualizar su ubicación o eliminarlo en caso de ser necesario.

Figura 38. Casos de uso del usuario

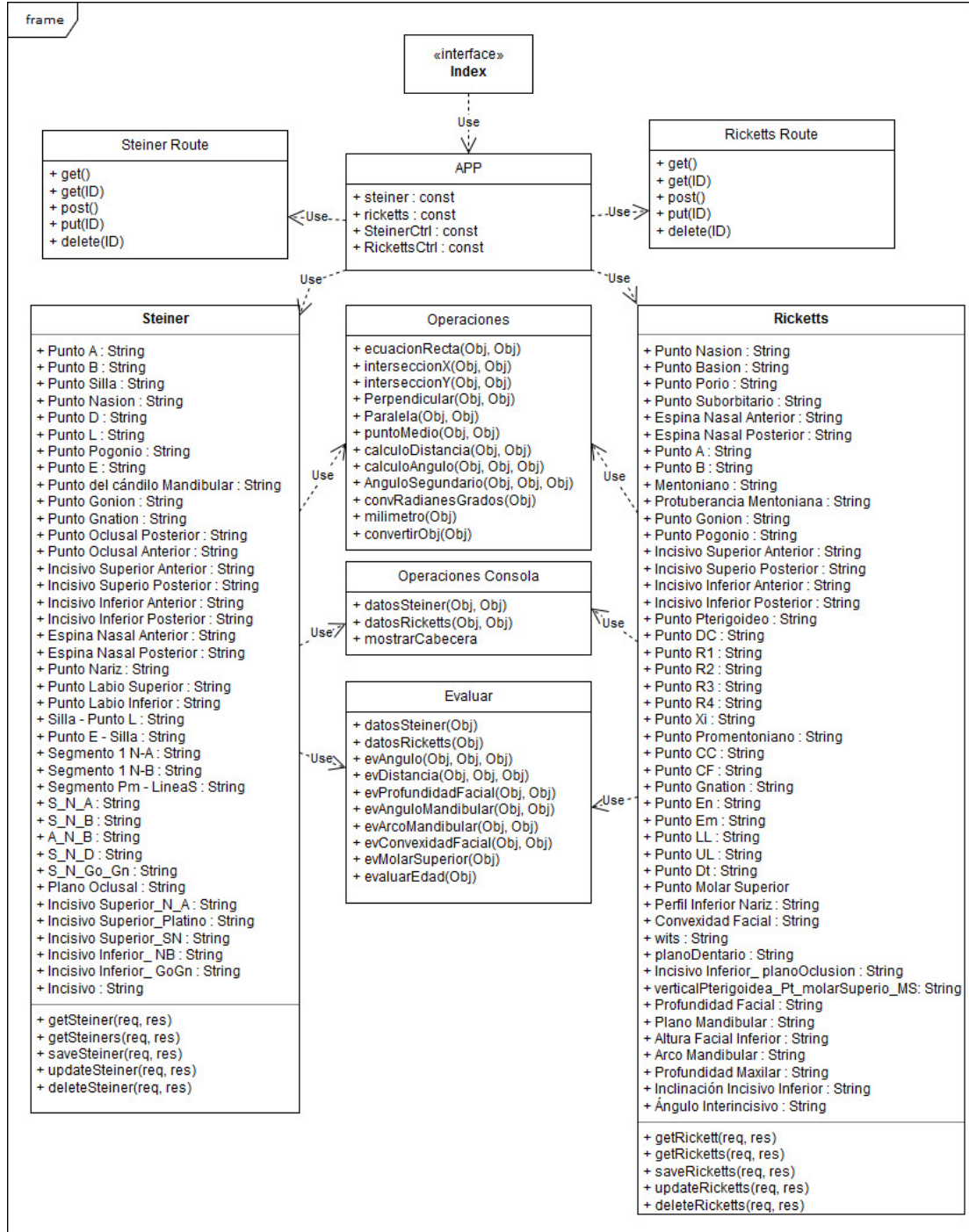


Fuente: González Luis (2017)

4.2.1.2. Diagrama de Clases

Gracias al diagrama de clases se tiene una mejor perspectiva de la estructura del software, de cómo funcionan sus clases y se relacionan entre sí, a que funciones puede acceder las cefalometrías y con qué se comunica a la interfaz desarrollada en la Fase I

Figura 39. Diagrama de Clases

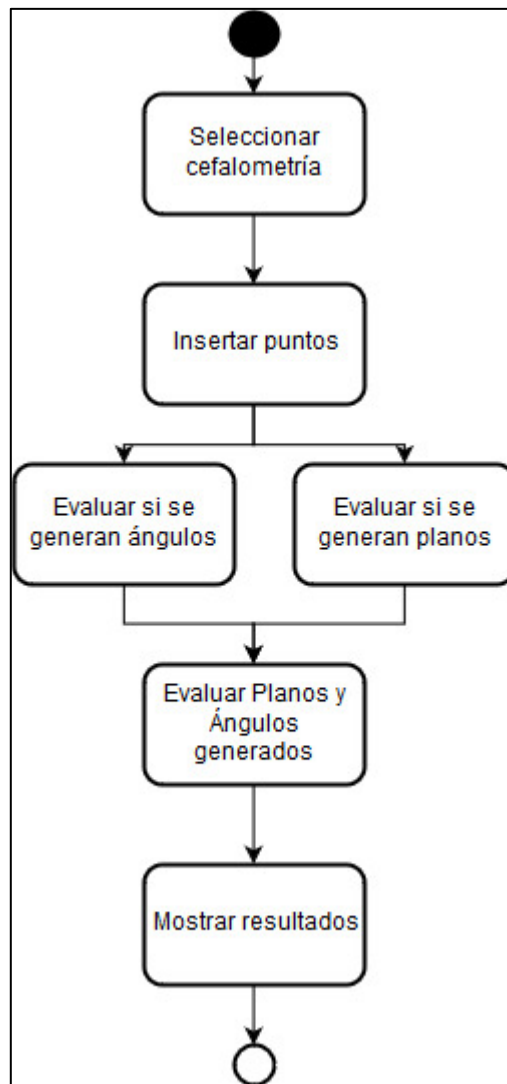


Fuente: González Luis (2017)

4.2.1.3. Diagrama de estados

En el diagrama de estados se mostrará cómo funciona la API, cuales son los estados por los que pasa, paso a paso hasta llegar a termina su ciclo.

Figura 40. Diagrama de estados



Fuente: González Luis (2017)

4.2.2. Glosario de términos

Se diseñó un glosario de términos para que sea fácil comprender lo escrito dentro del código de la API

Tabla 3. Glosario de términos

Nombre de Variable	Ruta	Función/Ruta	Uso
mongoose	index.js, models/steiner.js	Main	Para conexión con la base de datos
app	index.js	Main	Para hacer el llamado a la APP
config	index.js	Main	Para la configuración base de la API
express	app.js, routes/SteinerRoute.js, routes/RickettsRoute.js	Main	Facilita la creación de servidor y la llamadas HTTP
bodyParser	app.js	Main	Permite parsear los datos a un JSON

Nombre de Variable	Ruta	Función/Ruta	Uso
app	app.js	Main	Es a través de donde se usara la librería express
steiner	app.js	Main	Permite acceder a las rutas que permiten acceder o modificar datos de la cefalometría de Steiner
ricketts	app.js	Main	Permite acceder a las rutas que permiten acceder o modificar datos de la cefalometría de Ricketts
SteinerCtrl	app.js, routes/SteinerRoute.js	Main	Permite acceder al controlador de la cefalometría de Steiner, donde están sus

Nombre de Variable	Ruta	Función/Ruta	Uso
			diferentes métodos
RickettsCtrl	app.js, routes/RickettsRoute.js	Main	Permite acceder al controlador de la cefalometría de Ricketts, donde están sus diferentes métodos
steiner	routes/SteinerRoute.js	Main	Declaramos una variable donde se usarán las rutas HTTP a través de express
ricketts	routes/RickettsRout.js	Main	Declaramos una variable donde se usarán las rutas HTTP a través de express
Steiner	controllers/steiner.js	Main	Se hace el llamado al modelo de la

Nombre de Variable	Ruta	Función/Ruta	Uso
			cefalometría de Steiner
Ricketts	controllers/ricketts.js	Main	Se hace el llamado al modelo de la cefalometría de Ricketts
opc	controllers/steiner.js, controllers/ricketts.js	Main	Se usa para incluir las operaciones matemáticas necesarias para los diferentes análisis
ev	controllers/steiner.js, controllers/ricketts.js	Main	Se usa para enviar los datos obtenidos a evaluar, para ver si los resultados se encuentran dentro de los estándares

Nombre de Variable	Ruta	Función/Ruta	Uso
mostrarConsola	controllers/steiner.js, controllers/ricketts.js	Main	Se usa para una vez obtenido los datos de las cefalometrías se muestren de una manera amigable mediante la consola de Windows
steinerId	controllers/steiner.js	getSteiner(), updateSteiner(), deleteSteiner()	Variable usada para recibir el Id de la cefalometría que se esté trabajando
rickettsId	controllers/ricketts.js	getRicketts(), updateRicketts(), deleteRicketts()	Variable usada para recibir el Id de la cefalometría que se esté trabajando
steiner	controllers/steiner.js	saveSteiner(), updateSteiner()	Variable que se usará para

Nombre de Variable	Ruta	Función/Ruta	Uso
			guardar todos los posibles puntos o datos que se vayan generando durante el análisis
ricketts	controllers/ricketts.js	saveRicketts(), updateRicketts()	Variable que se usará para guardar todos los posibles puntos o datos que se vayan generando durante el análisis
Schema	models/steiner.js, models/Ricketts.js	Main	Variable que se usa para hacer uso de los esquemas que posee mongoose
SteinerSchema	models/steiner.js	Main	Variable utilizada para guardar los datos que se guardarán en la base de

Nombre de Variable	Ruta	Función/Ruta	Uso
			datos con respecto a la cefalometría de Steiner
RickettsSchema	models/ricketts.js	Main	Variable utilizada para guardar los datos que se guardarán en la base de datos con respecto a la cefalometría de Ricketts

Fuente: González Luis (2017)

4.2.3. Riesgos

Los riesgos se tomaron en consideración desde el primer momento y durante la planificación para que así el margen de error fuera nulo, inclusive se planifico teniendo en cuenta ciertos problemas como la complejidad del diseño del sistema y su estructura interna, limitaciones referentes al tiempo y recursos empleados para la realización exitosa de dicha aplicación (API-REST) que pudieran surgir.

4.2.4. Funcionabilidad extra

A pesar que el Software de trazado puede ir mejorando cada vez, añadiendo muchas funcionalidades nuevas, desde nuevos análisis cefalométricos hasta el uso de la API en diferentes idiomas, no se implementaron esas posibles opciones ya que no estaban en el planteamiento inicial, sin embargo, se diseñó la API de tal forma, que al querer añadir nuevas funcionalidades todo se diera de una manera sencilla gracias a la estructura que ésta posee.

4.2.5. Refactorizar

Se buscó siempre refactorizar el código a manera de optimizarlo lo más posible, reduciendo el tiempo de respuesta considerablemente y creando un cómodo método de intercambio de información con la interfaz del cliente.

4.3. Fase III – Codificación

La codificación se llevó a cabo de la mejor manera posible tomando en cuenta los diagramas diseñados previamente para así evitar confusiones con la interfaz del Software de trazado, para ello se desarrollaron varios módulos donde se encontraba la información específica de cada sección. Además de que todo se trabajó con un MVC (Modelo, Vista, Controlado) a pesar de solo ser la API del Software de trazado, se configuro de tal manera que al momento de realizar la integración de la interfaz con la API funcione sin ningún tipo de problema

4.4. Fase IV – Pruebas

Para las pruebas de la API del Software de trazado se usó un cuadro con fechas establecidas y en un entorno controlado para asegurarse de que toda la información que se esté suministrando sea verdadera, y se pueda evaluar la respuesta de la API cómodamente. Para estas pruebas se siguió el modelo planteado con anterioridad, y a continuación en la tabla 4 se muestran los resultados que se obtuvieron de cada prueba que se realizó.

Tabla 4. Resultados de pruebas

Prueba	Resultado
Comparar modelos de Sistemas con SQL y NoSQL	Se obtuvo como resultado que para el desarrollo del Software de trazado es mejor trabajar con NoSQL, por la velocidad de respuesta y la cantidad de peticiones que puede soportar
Prueba del Web Service	Se estableció un servidor local donde se comprobó el buen funcionamiento del web service enviado y recibiendo datos de manera controlada y asegurándose que la información devuelta sea la adecuada
Prueba de la API-REST	Una vez establecido el servidor local se probó que el software soportara ser trabajado como API- REST evaluando sus ventajas y desventajas para luego posteriormente probar el funcionamiento en el servidor local.
Simulación de solicitud de análisis de Steiner con la API-REST	Se comprobó que la API del software de trazado devolviera los datos precisos que se necesitaba en el análisis de Steiner, esto con la ayuda de datos comprobados de manera manual y bajo supervisión de alguien con experiencia de ese análisis cefalométrico
Simulación de solicitud de análisis de Ricketts con la API-REST	Una vez comprobado el correcto funcionamiento del análisis de Steiner fue

Prueba	Resultado
	más sencillo el análisis de Ricketts, ya que se tenía la información previa validada, solo se debía adaptar a los nuevos puntos que utilizaba Ricketts, los cuales generaban nuevos planos y ángulos

Fuente: González Luis (2017)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Luego de finalizar el desarrollo de la API que será parte del sistema del Software de Trazado se llegaron a las siguientes conclusiones:

- No se pudo analizar los datos obtenidos en la Fase I del Software de trazado ya que por falta de tiempo no se pudo concluir la elaboración de la interface, sin embargo, se estableció una estructura de datos que sin importar la interfaz que se quiera conectar con la API solo debían enviar los puntos del análisis cefalométrico seleccionado en coordenadas (X; Y) para que el sistema pueda analizar esos puntos y devolver los resultados en un JSON que puede ser interpretado fácilmente por cualquier software.
- La Fase II se inició sin contratiempos desarrollando los algoritmos necesarios para la resolución de los diferentes análisis que se hacen en las cefalometrías de Steiner y Ricketts, leyendo los puntos que llegan a través de la interfaz y analizándolos para su posterior exportación en formato JSON.
- La estructura que se utilizó permite la incorporación a nuevos análisis cefalométricos de una manera muy sencilla, ya que trabaja a través de un MVC sin las vistas, tendiendo el controlador y el modelo en JavaScript para que pueda ser interpretado por una gran variedad de interfaces.
- Se decidió desarrollar una API que fuera la encargada del análisis de los puntos obtenidos a partir de las imágenes de una radiografía panorámica. Esta API contaba con las estructuras para el análisis de las cefalometrías de Steiner y Ricketts, y con posibilidades de incluir nuevas cefalometrías de manera sencilla y estructurada.

- Estructurar la base de datos con NoSQL permite una muy rápida respuesta al cliente, tanto para almacenar datos como para consultarlos, en comparación con los sistemas con una base de datos con SQL.
- Las pruebas se dieron en un entorno controlado dentro de la Universidad José Antonio Páez con la ayuda del profesor de odontología Oswaldo Mejías, quien probó la API a través de una interfaz genérica (POSTMAN) obteniendo resultados favorables y como se esperaban.

Los resultados obtenidos durante cada fase metodológica, nos permitieron dar cumplimiento al objetivo general y los objetivos específicos de la mejor manera posible, y se pudo concluir que:

- Durante la primera fase de la investigación, en donde se aplicó la entrevista mixta como método de recolección de datos se pudo comprender la necesidad de la Universidad José Antonio Páez de crear un software de trazado para el análisis de radiografías cefálicas laterales, donde se aplican estudios como el análisis de Steiner y el análisis de Ricketts. En base a toda la información que se pudo obtener se elaboró el plan de desarrollo de la API que usaría el Software de trazado solicitado por la universidad.
- Para el diseño de la estructura que usaría la API los diagramas UML contribuyeron dando una mejor idea del cómo se relacionan las diferentes clases y cuáles son las secuencias por las que pasa la información, además de ver cuáles serían las funciones que cumplirían tanto el administrador como el usuario.
- El desarrollo de la aplicación se vio bastante influenciada por usar un sistema de almacenamiento NoSQL, donde la base de datos es no relacional, permitiendo una mejor respuesta por parte del servidor, sumando a una

estructura de código de JavaScript con Node.js para que pueda ser utilizado en diferentes plataformas sin complicaciones.

Para concluir, el desarrollo de esta API que formará parte del Software de trazado que utilizará la Universidad José Antonio Páez nos ayudó a reconocer la importancia de los conocimientos adquiridos dentro de la institución, tanto los técnicos como los prácticos al momento de poder dar solución a un problema que se presenta dentro de la casa de estudio llenando de satisfacción al autor de este trabajo.

5.2. Recomendaciones

Con el fin de mantener la API del software de trazado en correcto funcionamiento y ampliar el alcance de esta se conciben las siguientes recomendaciones:

- Realizar mantenimientos correctivos al equipo contenedor del sistema, para mantener el correcto funcionamiento del mismo y prevenir posibles pérdidas de información o retraso en el tiempo de respuesta.
- Seguir incluyendo análisis cefalométricos bajo la misma estructura que ya está establecida aprovechando todas las opciones que permite el sistema para evaluar los diferentes puntos ingresados
- Realizar respaldo de la información de la base de datos generada en los servidores.
- Conectar la API con interfaces que estén aptas para trabajar con API-REST para obtener así un resultado mucho más veloz.
- No limitar el uso de la API solo a la Universidad José Antonio Páez, ya que, así como ésta institución quería actualizar su manera de evaluar radiografías cefálicas laterales, otras casas de estudios pueden necesitarla.
- La API tiene licencia de copyleft de para que el código sea de libre distribución siempre y cuando los derechos de autor sean preservados.

BIBLIOGRAFÍA

- Álvares, C. (9 de Julio de 2014). *Genbetadev*. Recuperado el 11 de Enero de 2017, de Como funciona node js: <https://www.genbetadev.com/frameworks/como-funciona-node-js>
- Andrearrrs. (30 de Abril de 2014). *Un vistazo a Node.js: JavaScript en el servidor*. Recuperado el 11 de Marzo de 2017, de HiperTextual: <https://hipertextual.com/archivo/2014/04/nodejs-javascript-servidor/>
- Asier, M. (11 de Noviembre de 2013). *Conceptos sobre APIs REST*. Recuperado el 15 de Mayo de 2017, de Asier Marqués: <http://asiermarques.com/2013/conceptos-sobre-apis-rest/>
- BBVAOpen4U. (21 de Marzo de 2016). *APIs para el Internet de las Cosas: ThingSpeak, Pachube y Fitbit*. Recuperado el 15 de Mayo de 2017, de BBVA API_MARKET: <https://bbvaopen4u.com/es/actualidad/apis-para-el-internet-de-las-cosas-thingspeak-pachube-y-fitbit>
- Bruxita. (agosto de 2010). *Interpretación de Ricketts*. Recuperado el diciembre de 2016, de Bruxita: <https://bruxistas.files.wordpress.com/2010/09/interpretacion-ricketts.pdf>
- Bustamante, D., & Rodríguez, J. (2014). *Metodología Actual, Metodología XP*. Barinas.
- Clinica Pro dental SL. (2017). *Análisis Cefalométrico*. Recuperado el diciembre de 2016, de Clinica Pro dental SL: <https://www.pro dental.es/ortodoncia/analisis-cefalometrico/>
- Clínicas Pro dental SL. (2017). *Tipos de maloclusión*. Recuperado el diciembre de 2016, de Clínicas Pro dental SL: <https://www.pro dental.es/ortodoncia/maloclusion/>
- Clínicas Pro dental SL. (2017). *Tipos de radiografía dental*. Recuperado el noviembre de 2016, de Clínicas Pro dental SL: <https://www.pro dental.es/blog/odontologia/tipos-de-radiografia-dental/>
- Colgate. (2017). *¿Qué es la Ortodoncia?* Recuperado el enero de 2017, de Colgate: <http://www.colgate.com.ve/es/ve/oc/oral-health/life-stages/teen-oral-care/article/what-is-orthodontics>

- Cortés, F. (mayo de 2007). *Skeletal dysplasias*. Recuperado el diciembre de 2016, de Medware:
<http://www.medwave.cl/link.cgi/Medwave/PuestaDia/Congresos/817>.
- Express. (2017). *Express*. Obtenido de Express: <http://expressjs.com/es/>
- García, C., Morales, O., Oreallana, N., Ramírez, R., & Setién, V. (2008). Proyecto factible empleando el software recompX®. *La hipermedia y la enseñanza-aprendizaje de la odontología*. 46, págs. 1-13. Caracas: Acta Odontológica Venezolana. Recuperado el enero de 2017, de <http://www.saber.ula.ve/bitstream/123456789/27867/1/hipermedia.pdf>
- Java. (2017). *Conozca más sobre Java*. Recuperado el enero de 2017, de Java: <https://www.java.com/es/about/>.
- M. B., X. M., S. C., B. T., N. Z., & J. G. (2015). Estudio comparativo entre las superposiciones cefalométricas trazadas manual y digitalmente con Dolphin y Nemoceph. *Revista Española de Ortodoncia*, 45(3), 142-149.
- MedlinePlus. (2016). *MedlinePlus*. Recuperado el noviembre de 2016, de Radiografías dentales: <https://medlineplus.gov/spanish/ency/article/003801.htm>
- MongoDB, I. (2017). *MongoDB*. Recuperado el 11 de Marzo de 2017, de MongoDB: <https://www.mongodb.com/es>
- Network, M. D. (2017). *JavaScript, Mozilla Developer Network*. Recuperado el 11 de Marzo de 2017, de Mozilla Developer Network: <https://developer.mozilla.org/es/docs/Learn/JavaScript>
- Node.js. (2017). *Documentation*. Recuperado el 11 de Enero de 2017, de Node.js.
- Oracle Corporation. (2017). *La base de datos de código abierto más popular del mundo*. Recuperado el enero de 2017, de Oracle Corporation: <https://www.oracle.com/lad/mysql/index.html>
- Oracle Corporation. (2017). *NetBeans*. Recuperado el enero de 2017, de Oracle Corporation: <https://netbeans.org/features/index.html>.
- Orozco, L. (3 de febrero de 2015). *Cefalometría de Steiner*. Recuperado el diciembre de 2016, de Prezi: <https://prezi.com/jewoarhd2alz/cefalometria-de-steiner/>
- PHP. (2017). *¿Qué es PHP?* Recuperado el enero de 2017, de PHP: <http://php.net/manual/es/intro-whatis.php>
- PHP. (2017). *¿Qué puede hacer PHP?* Recuperado el enero de 2017, de PHP: <http://php.net/manual/es/intro-whatcando.php>

- Tamayo, T. y., & Mario. (1997). *El Proceso de la Investigación científica*. Mexico: Editorial Limusa S. A. Obtenido de Tesis de Investigadores : <http://tesisdeinvestig.blogspot.com/2011/06/poblacion-y-muestra-tamayo-y-tamayo.html>
- Universidad de Chile, Facultad de Odontología. (2010). *Puntos cefalométricos a nivel de la base del cráneo*. Santiago de Chile.
- Universidad de Costa Rica, Facultad de Odontología. (2006). *Principales análisis cefalométricos utilizados para el diagnóstico ortodóntico*. Costa Rica.
- Universidad Federal de la Integración Latinoamericana, Fundación Herminio Ometto. (2010). *Análisis cefalométrico resumido de Ricketts*. Brasil.
- Wikipedia. (2 de Enero de 2017). *Servicio Web*. Recuperado el enero de 2017, de Wikipedia: https://es.wikipedia.org/wiki/Servicio_web
- Wikipedia. (23 de Mayo de 2017). *Transferencia de Estado Representacional*. Recuperado el 15 de Mayo de 2017, de Wikipedia: https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional

ANEXOS

Anexo A: Cronograma de actividades

Actividad	Mes							
	Noviembre	Diciembre	Enero	Febrero	Marzo	Abril	Mayo	Junio
Diagnosticar y planificar el desarrollo del software de trazado.	X	X	X					
Diseñar la estructura interna que utilizará la API en el software de trazado .			X	X				
Desarrollar el software de trazado que permita la interpretación de punto cefalométricos enviados desde la Fase I.				X	X	X	X	
Realizar las pruebas pertinentes para determinar el correcto funcionamiento de la API del software de trazado.							X	X

Fuente: González Luis (2017)

Anexo C: Modelo de entrevista

#	Pregunta
1	¿Qué es un software de trazado?
2	¿Conoce algún software de trazado en el mercado actual?
3	¿Cuáles análisis cefalométricos debe realizar un software de trazado comercial actualmente?
4	¿Cuáles análisis cefalométricos debe realizar el software de trazado que se realizará a la Universidad José Antonio Páez?
5	¿Conoce otras instituciones donde se pueda utilizar el software de trazado desarrollado para la Universidad José Antonio Páez?
6	¿Esas instituciones usan los mismos análisis cefalométricos que se utilizan dentro de la Universidad José Antonio Páez?
7	¿Quiénes usarían el software de trazado dentro de la Universidad José Antonio Páez?
8	¿Quiénes considera usted que deberían utilizar el software de trazado al momento de realizar las pruebas?

Fuente: González Luis (2017)