UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

HERRAMIENTA PARA LA ESTIMACIÓN DE PROYECTOS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

TOMÁS ISMAEL VERDUGO ZAMBRA

PROFESORA GUÍA:
MARÍA CECILIA BASTARRICA PIÑEYRO

MIEMBROS DE LA COMISIÓN:
ALEXANDRE BERGEL
ADOLFO CARRASCO A.

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE: INGENIERO CIVIL EN COMPUTACIÓN
POR: TOMÁS ISMAEL VERDUGO ZAMBRA
FECHA: 2022
PROF. GUÍA: CECILIA BASTARRICA PIÑEYRO

HERRAMIENTA PARA LA ESTIMACIÓN DE PROYECTOS ORIENTADO A PEQUEÑAS Y MEDIANAS EMPRESAS

En esta memoria se propone una herramienta para la estimación de proyectos de software, orientado a pequeñas y medianas empresas. La técnica implementada en la herramienta es una versión simplificada del Analytical Hierarchy Process, o AHP. La elección de la técnica para la herramienta nace como el resultado del análisis de varias técnicas, de las cuales la que más se beneficia por ser usada con una herramienta es el AHP. Las simplificaciones de la técnica hacen posible su uso reutilizando información en la herramienta, disminuyendo el trabajo manual que debe hacer un usuario para generar una estimación. Las simplificaciones también permiten usar datos cuantitativos de proyectos para generar la estimación. Se comenta la implementación de estas simplificaciones y se muestran casos de usos de la herramienta comunes en la creación de una estimación para un proyecto. Se procede también a analizar experimentalmente el comportamiento de la técnica simplificada con la base de datos finlandesa de proyectos, y se concluye que la técnica AHP en su estado original no puede hacer buen uso de los datos cuantitativos de proyectos.

Tabla de contenido

1. Introducción **1**
1.1. Técnicas de estimación . . . 1
1.2. AHP . . . 2
1.3. Herramientas para la estimación de proyectos . . . 3
1.4. Objetivo . . . 4
1.5. Estructura del documento de tesis . . . 4

2. Técnicas de estimación **5**
2.1. Estimeeting . . . 5
2.2. Wideband Dephi . . . 5
2.3. Planning Poker . . . 6
2.4. Planning Game . . . 6
2.5. Expert Judgment . . . 6
2.6. Guesstimation . . . 7
2.7. Work Breakdown Structure (WBS) . . . 7
2.8. Constructive Agile Estimation Algorithm (CAEA) . . . 7
2.9. Collaborative Filtering . . . 7
2.10. Analytical Hierarchical Process (AHP) . . . 7
2.11. Use Case Points (UCP) . . . 8
2.12. Tabla de características de las técnicas de estimación . . . 9
2.13. Elección de la técnica a implementar . . . 10

3. AHP: técnica base **11**
3.1. Métricas . . . 11
3.2. Proyectos . . . 11
3.3. La Jerarquía . . . 11
3.4. Ejemplo . . . 12
3.5. Comparaciones . . . 13
3.6. Relación inversa entre una métrica de la jerarquía con la métrica estimada . . . 14
3.7. Escala de valores para las comparaciones . . . 14

4. AHP simplificado: modificaciones a la técnica **15**
4.1. Clausura de comparaciones . . . 15
4.2. Comparaciones relajadas . . . 17
4.3. Uso de valores reales . . . 17
4.4. Reutilización de información . . . 17

5. Implementación de la técnica AHP Simplificada **19**
5.1. Jerarquías y métricas . . . 19
5.2. Tipos de nodo . . . 19
5.3. Comparaciones . . . 19
5.4. Computación de una clausura . . . 20
5.5. Generación de la matriz de comparaciones . . . 21
5.6. Computación aproximada de los eigenvectores de la matriz . . . 21
5.7. Propagación de resultados avanzando en la jerarquía . . . 21
5.8. Requisitos mínimos para la generación de una estimación . . . 22

5.9. Etapas en la generación de una estimación 22

6. Herramienta **24**
6.1. Arquitectura 24
6.2. Casos de uso 24

7. Análisis Experimental **60**
7.1. Mínimo error posible para las estimaciones 60
7.2. Jerarquía simple 60
7.3. Resolución automática de la jerarquía 63
7.4. Base de datos utilizada, Finnish Software Effort Dataset 64
7.5. Variación de proyectos 65
7.6. Variación de métricas 66
7.7. Uso de la jerarquía automática en proyectos no ajustados 70

8. Discusión **71**
8.1. Contexto ideal para el uso de la técnica 71
8.2. Limitaciones de la técnica 71
8.3. Mejoras a la técnica 71
8.4. Mejoras a la herramienta 71
8.5. Otros 72

9. Conclusión **74**

Bibliografía **75**

Índice de ilustraciones

1. Estimación de la jerarquía según el valor de comparación, con y sin escala. . 63
2. Error porcentual de la estimación según la cantidad de proyectos, con y sin escala. 65
3. Error porcentual de la estimación según la cantidad de proyectos, para jerarquías de 1, 2, 3, 5, 10 y 15 métricas. 66
4. error porcentual de la estimación de una jerarquía para cada métrica 67
5. error porcentual de la estimación de una jerarquía según la cantidad de métricas utilizadas. 68
6. error porcentual de la estimación para jerarquías de cada par de métricas. . . 68
7. error promedio para las jerarquías de cada par de métricas. 69
8. error promedio para las jerarquías evaluadas en otros proyectos, para cada par de métricas. 70

1. Introducción

Desde su concepción, los proyectos de software se benefician de estimaciones certeras. Las estimaciones ayudan a definir la asignación de recursos en estos proyectos, y su fallo suele significar pérdidas financieras [18]. A pesar de la importancia de las estimaciones, solo un cuarenta por ciento de los proyectos que se llevan a cabo se pueden considerar como exitosos [7], logrando cumplir con sus expectativas de costo, tiempo y satisfacción de los usuarios. Este fallo que se delata en la exactitud de las estimaciones existe a pesar de la existencia de abundantes herramientas de software [19] que permiten el uso de técnicas para la estimación requerida para llevar a cabo un proyecto de software. Tomando en cuenta el bajo nivel de éxito, junto a un supuesto elevado número de herramientas, llegamos a dos posibles escenarios: o las herramientas no tienen una llegada considerable en la industria del software, o si tienen llegada, las técnicas que implementan no se desempeñan lo suficientemente bien.

Hablamos de estimaciones en general pues estas suelen ser de costos, de tamaño o de tiempo, dependiendo del objetivo que se tiene al momento de hacer la estimación. Las técnicas de estimación suelen servir tanto para un tipo de estimación como para otro, algunas requieren de datos extras para convertir la estimación de un tipo (de tamaño por ejemplo) a otro (a tiempo). A lo largo del documento, cuando hablamos de estimaciones, hablamos de estimaciones en general sin diferenciar entre los tipos.

1.1. Técnicas de estimación

Respecto a las técnicas que se suelen utilizar para el cálculo de estimación en proyectos de software, tres de las técnicas utilizadas más populares son Function Points, COCOMO y Wideband Delphi [3]. En general, la mayoría de las técnicas que se utilizan son algorítmicas, las cuales suelen requerir información cuantitativa del proyecto para generar sus estimaciones [18], tipo de información que en las primeras etapas de un proyecto no es abundante.

Cabe señalar que se han hecho estudios para conocer si las técnicas algorítmicas, de las cuales hay una mayoría en el mercado, son superiores a las técnicas basadas en analogía, y los resultados indican que en efecto las técnicas algorítmicas se desempeñan mejor. También se han llevado a cabo estudios para demostrar lo contrario, que las técnicas basadas en analogía son superiores a las algorítmicas, y los resultados indican que en efecto las técnicas basadas en analogía se desempeñan mejor. Actualmente se cree que esta dicotomía en la capacidad de las técnicas para servir como herramientas de estimación depende de los datos utilizados en los distintos estudios, y que no existe una superioridad absoluta de un tipo de técnica sobre otra [9].

Respecto al alcance en la industria de las herramientas, se realizó una búsqueda de herramientas publicitadas como herramientas para hacer estimaciones de proyectos de software, y se encontró un número reducido de herramientas de las cuales el mayor número era pagada. También se buscó un par de herramientas generalmente mencionadas en la literatura (como archANGEL), pero estas no están disponibles a la fecha para su uso libre.

Factores que influyen en la adopción de herramientas en la industria es la disponibilidad de recursos de las compañías y la cantidad de datos propios disponibles para alimentar estas herramientas. Muchas de la empresas que existen son pequeñas o medianas y no tienen estos

recursos ni tienen estos datos necesarios para hacer uso de estas herramientas [24].

Con pocas herramientas de uso libre, y con un requisito de recursos y datos elevado para el uso de las herramientas en general, la industria no favorece a las pequeñas y medianas empresas. Para ayudar a estas pequeñas y medianas empresas a adoptar y utilizar herramientas de estimación, las herramientas se tienen que adaptar al perfil de estas empresas.

Esta memoria propone una herramienta que se ajusta a este perfil. Una herramienta que no se basa fuertemente en datos y de uso libre. El objetivo de esta memoria es la construcción de dicha herramienta, una herramienta de apoyo para la estimación de proyectos en el desarrollo de software orientado a las pequeñas y medianas empresas.

Para lograr este objetivo, la técnica que implementa esta herramienta es una versión del Analytic Hierarchy Process (AHP).

1.2. AHP

La elección de la técnica principal a implementar, el Analytic Hierarchy Process (AHP), surge como resultado del uso del catálogo de técnicas que ofrece [24], así como de la caracterización de pequeñas y medianas empresas y de las técnicas que más adecuadas son para estas. Estas técnicas son tratadas en la sección técnicas de estimación. La elección de la técnica AHP también es tratada en la sección técnicas de estimación.

El uso del AHP como herramienta para la estimación en proyectos de software se describe en [20], documento en el cual además se describen casos comparativos donde el desempeño de esta técnica logra mejores resultados que la estimación de experto, y que la estimación de una de las técnicas algorítmicas, basada en regresión.

Si bien la técnica AHP es ampliamente usada en varios campos industriales, comerciales y de investigación y es resistente ante el error humano, en el contexto de la estimación de proyectos de software requiere de un esfuerzo cuadrático respecto a la cantidad de proyectos que se utilicen para calcular la estimación del proyecto en cuestión (ya que cada nuevo proyecto considerado tiene que ser comparado con cada otro proyecto anteriormente considerado), lo que la vuelve trabajosa y poco atractiva en relación a las demás técnicas.

Abordamos este sobre esfuerzo que presenta AHP respecto a otras técnicas simplificando su uso al explotar la persistencia de datos, y el hecho de que mucha información utilizada en el cálculo de la estimación de un proyecto es la misma para la estimación de cualquier otro proyecto. Abordamos también este sobre esfuerzo dando al usuario la opción de cambiar resistencia y precisión de los resultados a cambio de ingresar una menor cantidad de información.

Recientemente se hizo un extenso análisis de las capacidades que posee la técnica AHP en el ámbito de la toma de decisiones [13], donde queda demostrado que no es una técnica apta para todo tipo de situación. Por esto mismo nos proponemos estudiar el desempeño para ver cómo varía según el nivel de información que incluya el usuario para generar las estimaciones.

1.3. Herramientas para la estimación de proyectos

Como esta memoria propone una herramienta para la estimación de proyectos, nos interesa saber del ecosistema de las herramientas para la estimación de proyectos de software.

Herramientas de uso abierto

- Existe una plétora de herramientas que implementan planning poker: plataformas web, aplicaciones android y power-ups para trello por nombrar unos pocos formatos.
- Estimatool [6], una herramienta que se basa en una técnica híbrida entre los puntos de función y los puntos de casos de uso (UCP).
- Orange Effort Estimation Tool [15] es una herramienta abierta para la descarga que implementa varias técnicas: COCOMO II, Work Breakdown Estimation, Analogy / Comparison Estimation y Custom modular estimation for WEB and Mobile.

Herramientas pagadas

A continuación se da una lista de herramientas pagadas.

Herramientas que utilizan la técnica de puntos de función (puede que algunas de estas herramientas den la opción de utilizar más técnicas y no solo puntos de función, pero esta información es desconocida).

- Productivity Data Query Tool
- SEER by Galorath
- Experience Services
- SCOPE Project Sizing Software (de Totalmetrics)
- TruePlanning (PRICE)
- PRICE Cost Analitics (proprietary)
- WORKBENCH
- ScopeMaster

Herramientas que usan técnicas de minería de datos.

- BlueOptima
- QSM SLIM-Estimate

Herramientas inalcanzables

A continuación se listan herramientas mencionadas en la literatura, herramientas a las que no se tiene acceso, algunas por enlaces caídos, otras por acceso restringido.

- LEAP [8]
- DataSalvage [20]
- ANGEL (o archANGEL) [10]
- Awesome Estimator [1], una herramienta que se basa en la técnica CEPF-I [16].

1.4. Objetivo

El objetivo de esta memoria es la construcción de una herramienta de apoyo para la estimación de proyectos en el desarrollo de software orientado a las pequeñas y medianas empresas. Que no se base fuertemente en datos y que sea de uso libre.

Para lograr este objetivo, se definen los siguientes objetivos específicos:

- Implementar una versión simplificada del Analytic Hierarchy Process (AHP) que disminuya el esfuerzo humano requerido para el uso de la técnica.
- Tener una interfaz de usuario simple e intuitiva orientada a la expresividad gráfica para una fácil adopción de la herramienta.
- Hacer un diseño extensible para facilitar la integración de futuras técnicas para el cálculo de estimaciones.

La evaluación de la técnica AHP simplificada se medirá utilizando una base de datos de proyectos que se conseguirá para este propósito. La evaluación consistirá en la variación de la cantidad de información que ingresa el usuario para generar una estimación en la técnica AHP simplificada, permitiendo saber así el nivel de certeza que entregan estas estimaciones.

1.5. Estructura del documento de tesis

La siguiente sección trata de las diferentes técnicas de estimación consideradas como candidatas para su implementación en la herramienta, terminando con la lógica detrás de la elección de la técnica AHP como técnica base para ser simplificada e implementada. La sección tres de esta memoria trata de la técnica AHP vista a la luz de sus partes relevantes para la implementación.

La sección cuatro trata la base teórica de la simplificación de la técnica AHP. La sección cinco trata de la implementación en la herramienta de la técnica AHP simplificada, con un enfoque en los algoritmos utilizados. La sección seis trata del uso de la herramienta, ejemplificando su uso con casos de uso.

La sección siete trata del análisis experimental de la técnica AHP simplificada. En la sección ocho se discuten varias ideas, como las limitaciones de la técnica AHP simplificada, el contexto ideal para el uso de la técnica AHP simplificada, y varias mejoras para la técnica, la implementación y la herramienta. Y terminando, la sección nueve trata las conclusiones de la memoria como contraste a los objetivos.

2. Técnicas de estimación

Para encontrar las técnicas ideales a implementar en una herramienta para las pequeñas empresas nos centramos en las recomendadas en [24].

1. Estimeeting [22]
2. Wideband Delphi* [3]
3. Planning Poker [3]
4. Planning Game [21]
5. Expert Judgment [12]
6. Guesstimation
7. Work Breakdown Structure (WBS) [23]
8. Constructive Agile Estimation Algorithm (CAEA) [2]
9. Collaborative Filtering [14]
10. Analytic Hierarchy Process (AHP) [17], [20], [11]
11. Use Case Points (UCP) [5]

*La lista original incluye el método UCP Method Modification y en vez incluir el método wideband delphi, incluye el método delphi. La omisión se hace por la falta de información respecto a esta versión del UCP, y la diferencia entre la técnica dephi y la técnica wideband delphi es la etapa de discusión que incluye la versión considerada en este documento.

La caracterización hecha sobre las pequeñas empresas usadas para crear esta lista de técnicas es la siguiente: se asume que una empresa pequeña tiene poco o nada de datos respecto a proyectos anteriores, que usan técnicas basadas en el conocimiento experto y que utilizan técnicas no propietarias.

A continuación se describen estas técnicas, se explica su funcionamiento y otras características relevantes. Una descripción más detallada se puede encontrar en el Anexo 1.

2.1. Estimeeting

Descripción Es una meta estimación donde un grupo de personas revisa, seleccionando y aprobando las funcionalidades definidas para un software, así como las estimaciones asociadas a cada funcionalidad, dando como resultado las funcionalidades finales, así como las estimaciones finales para cada funcionalidad.

Proceso Para cada funcionalidad, un equipo hace un análisis de la funcionalidad, y una estimación que luego es revisada y juzgada por un grupo de personas en la estimeeting. Cada estimeeting revisa entre dos y cuatro funcionalidades.

2.2. Wideband Dephi

Descripción Es una meta estimación donde se comparan varias estimaciones hechas con la estimación base. La comparación se lleva a cabo en una discusión estructurada entre un

grupo de personas, y la estimación que se suele utilizar de base es el WBS (Work Breakdown Structure).

Proceso El proceso consiste en que se elige un grupo de personas para llevar a cabo la estimación, y cada persona del grupo hace (o modifica) una estimación base. Luego discuten los resultados obtenidos, haciendo énfasis en las estimaciones más alta y más baja. Se crea una estimación final a partir de las estimaciones del paso anterior, y por último se revisa esta estimación, considerándose su mejora y/o actualización.

2.3. Planning Poker

Descripción Es una técnica utilizada en proyectos de estructura ágil. Su principal uso es la estimación del tiempo o esfuerzo requerido para completar historias de usuario. Estimaciones de este tipo se realizan en cada iteración del proyecto ágil. Es una estimación grupal que compara estimaciones personales como en wideband delphi. Se diferencia de esta anterior en que las estimaciones bases son valores de un grupo discreto, y no estimaciones en sí.

Proceso Cada participante de un equipo de personas hace una estimación del proyecto por su cuenta, usando valores de un set (números de la serie de fibonacci por ejemplo) como posibles valores para la estimación. Luego el equipo discute estos valores, haciendo énfasis en los valores máximo y mínimo. Este proceso de hacer estimaciones personales y discutirlas se repite hasta alcanzar un consenso, o hasta concluir la insuficiencia de información para alcanzar uno.

2.4. Planning Game

Descripción Es una técnica utilizada en extreme programming que busca crear un ordenamiento de importancia de historias de usuario a ser implementadas en un proyecto o iteración. Una de las etapas de esta metodología consiste en hacer estimaciones del tiempo requerido para implementar dichas historias de usuario.

Proceso La etapa asociada a la estimación de las historias de usuario consiste en crear estimaciones del tiempo requerido para implementar estas historias de usuario, y clasificarlas según el nivel de confianza que se tiene en estas estimaciones considerando 3 niveles: una estimación certera, una estimación suficientemente buena, y una imposibilidad de hacer una estimación.

2.5. Expert Judgment

Descripción Hace referencia a las estimaciones que lleva a cabo un experto, basándose en su conocimiento de proyectos anteriores y su dominio del tema del proyecto. Su precisión depende de los conocimientos y del razonamiento del experto.

Proceso No tiene un proceso fijo.

2.6. Guesstimation

Descripción Es una estimación hecha sin tener al alcance información suficiente para validarla. Similar a una estimación de experto, solo que esta no cuenta con el análisis que el experto lleva a cabo para crear su estimación.

Proceso No tiene un proceso fijo.

2.7. Work Breakdown Structure (WBS)

Descripción Es una planificación del proyecto que lo divide en partes de tamaños manejables.

Proceso Se divide el proyecto en partes, y luego estas en partes se vuelven a dividir en partes más pequeñas hasta llegar a una granularidad de una (o dos) semanas de trabajo necesario para terminar una parte, llamada milestone. La estimación final es la agregación de la estimación del tiempo requerido para cada milestone.

2.8. Constructive Agile Estimation Algorithm (CAEA)

Descripción CAEA es un una técnica para proyectos ágiles que ocupen puntos de historia (story points). Consiste en el cálculo de un factor de corrección para los story points que se generan en las iteraciones de una metodología ágil.

Proceso Para el cálculo del factor de corrección se calculan los factores vitales del proyecto (como la seguridad requerida y el volumen de transacciones de datos), y se ponderan los story points por un factor calculado como la suma de los grados de cada factor vital, utilizando una escala a elección (como la serie de Fibonacci).

2.9. Collaborative Filtering

Descripción Permite calcular el esfuerzo requerido por un proyecto de software al hacer una comparación de las métricas del proyecto frente a las métricas de otros proyectos. Está diseñado para trabajar con una base de datos de proyectos con métricas dispares, algunos proyectos con unas métricas y otros con otras métricas.

Proceso Una vez se tienen estimados los valores de métricas relevantes para el proyecto a estimar, el algoritmo consulta la base de datos en busca de proyectos similares. Un proyecto es similar si comparte métricas con el proyecto a estimar, y su nivel de similitud se basa en el número de métricas compartidas, y en los valores similares de estas métricas. La estimación final se logra al agregar los datos de tiempo (o esfuerzo) de los proyectos similares, ponderados por su nivel de similitud.

2.10. Analytical Hierarchical Process (AHP)

Descripción La técnica consiste en dividir el proyecto en una jerarquía de características, y una serie de comparaciones por característica del proyecto con proyectos anteriores. Las

comparaciones usan una escala de una sola magnitud, para mantener las comparaciones relevantes. Requiere de un solo punto de referencia para funcionar (un solo proyecto con datos reales).

Proceso Se comparan todos los pares de proyectos en el nivel más bajo de la jerarquía, y como resultado se consigue una matriz de comparaciones, de la cual se calculan luego los pesos que influyen sobre los elementos superiores de la jerarquía. Para cada métrica, los proyectos se comparan de a dos, definiendo su relación de tamaño en una escala de 1 a 9 (e.g. uno es 5 veces más grande que el otro). Si en una métrica, los proyectos se diferencian mucho, entonces se agrupan los proyectos pequeños y las comparaciones se hacen dentro del grupo, y luego el grupo se compara con los demás proyectos grandes. De la misma manera en la que se define la relación de tamaño entre proyectos, se define la relación de peso entre métricas. Una vez se tiene el peso de las métricas se pasa a propagar los pesos de los proyectos y a agregarlos hasta conseguir un único valor para cada proyecto. Este único valor corresponde al nivel superior de la jerarquía y es la estimación calculada.

2.11. Use Case Points (UCP)

Descripción Es una estimación de tamaño para proyectos que se lleven con la metodología de use cases. Esta estimación se basa en el conteo y/o graduación de características bien definidas del proyecto. Estas graduaciones utilizan escalas de tres o seis valores posibles, haciendo el uso de esta técnica un acto simple y rápido.

Proceso Consiste en la resolución de una ecuación de cuatro partes, una parte que considera el número de casos de uso, una parte que considera el número de actores (usuarios u otras aplicaciones), una parte que considera el aspecto técnico del proyecto, y una parte que considera el contexto del proyecto. Cada caso de uso y cada actor se clasifica en una escala con tres valores (simple, normal y complejo) y cada característica técnica y cada característica contextual se clasifica en una escala de 0 a 5, con cero significando ninguna importancia, uno siendo la menor importancia y 5 siendo la mayor importancia. Luego estos valores se agregan para generar un valor que representa el tamaño del proyecto.

2.12. Tabla de características de las técnicas de estimación

A continuación se presenta una tabla de características de las técnicas anteriormente descritas. Esta tabla guía la elección de la técnica a ser implementada por la herramienta propuesta por esta memoria, elección presente en la sección de solución propuesta.

Nombre Técnica	De Grupo	Apoyada por una computación	Útil en etapa iterativa Ágil	Útil en etapa inicial del proyecto	Metodología
Estimeeting	sí	no	no	sí	discusión
Wideband Delphi	sí	no	sí	sí	discusión
Planning Poker	sí	no	sí	no	discusión
Planning Game	sí	no	sí	sí	variado
Expert Judgment	no	no	sí	sí	variado
Guesstimation	no	no	sí	sí	variado
Work Breakdown Structure	no	no (para calcular presición sí)	sí	sí	análisis descriptivo
CAEA	no	sí	sí	no	formulario
Collaborative Filtering	no	sí	con datos por historia de usuario	sí	analogía / formulario
AHP	no	sí	con datos por historia de usuario	sí	analogía / formulario
UCP	no	sí	no	en proyecto UC	formulario

- variado: puede ser cualquier metodología
- formulario: se asignan valores a una lista de características
- analogía: proyecto estimado comparado con otros proyectos

De la tabla podemos ver que de las técnicas consideradas, tanto Planning Poker como CAEA no se aplican al inicio de un proyecto, debido a que se aplican únicamente al inicio de cada iteración de un proyecto de metodología ágil. Vemos también que Estimeeting, Wideband Delphi, Planning Poker y Planning Game son todas técnicas que requieren un grupo de personas para ser utilizadas. Vemos también que CAEA, Collaborative Filtering, AHP y UCP se apoyan en algún tipo de computación para el cálculo de su estimación (y para el cálculo de la precisión de la estimación en el caso de WBS).

Cabe mencionar que debido a la naturaleza de analogía de collaborative filtering y de AHP, además de poder usarse para la estimación de proyectos, estas técnicas se pueden usar para

la estimación de las historias de usuario, permitiendo su uso en iteraciones de metodologías ágiles.

2.13. Elección de la técnica a implementar

Todas las técnicas consideradas se pueden beneficiar de un software para la persistencia de las estimaciones, como su ordenamiento y fácil acceso. Nos centraremos pues en el beneficio que trae una herramienta de software a la técnica obviando el punto anterior, esto es, aquellas técnicas que requieren de un cómputo: CAEA, Collaborative Filtering, AHP, UCP y WBS. Una de estas técnicas implementaremos, aquella que mejor aproveche los beneficios de una herramienta, y que a la vez se ajuste mejor al perfil de las pequeñas y medianas empresas.

En el caso de WBS solo el cálculo de la precisión se beneficia de un cómputo y no el cálculo de la estimación en sí. Tanto CAEA como UCP son técnicas que sólo se usan en ciertos tipos de proyectos, CAEA en proyectos ágiles y UCP en proyectos de casos de uso. Y Collaborative Filtering y AHP son técnicas que sirven en una gama más amplia de proyectos. Y de entre estas dos, Collaborative filtering se basa más fuertemente en datos que AHP, y como buscamos una técnica que requiera la menor cantidad de datos como parte de nuestras consideraciones para las pequeñas y medianas empresas, y como AHP requiere a lo menos un solo punto de referencia (un solo proyecto con el dato de su duración para la estimación de tiempo, por ejemplo), nos centramos en la implementación de la técnica AHP. Esto no significa que no se implementaran otras técnicas (que como lo plantea uno de los objetivos, se espera tener más de una técnica disponible en la herramienta), esto significa que el esfuerzo principal puesto a esta herramienta será puesto en el desarrollo de la técnica AHP.

3. AHP: técnica base

La técnica AHP consiste en comparar proyectos bajo varias métricas, y luego comparar estas métricas para generar una estimación de una métrica para uno de estos proyectos.

Las tres componentes principales de la técnica AHP son las Métricas, los Proyectos y la Jerarquía.

3.1. Métricas

Las métricas son propiedades de los proyectos. Propiedades como el costo, el tamaño del equipo de desarrollo, la calidad de las herramientas de desarrollo utilizadas por el equipo, etc.

La técnica AHP permite la estimación de cualquiera de estas métricas. En el contexto de la estimación de proyectos, se suele estimar las propiedades como el tiempo de desarrollo requerido, o un índice de esfuerzo del proyecto para llevar una buena planificación o para la negociación de un contrato.

Como se explica en [17], con esta técnica se pueden hacer estimaciones de esfuerzo para un proyecto a partir de su comparación con otros proyectos, bajo esta misma métrica de esfuerzo. También se explica como se pueden hacer estimaciones de esfuerzo a partir de otras métricas (que no son esfuerzo). En el caso general, esta técnica permite hacer estimaciones de una métrica a partir de un conjunto de métricas, conjunto que puede o no incluir la métrica siendo estimada.

3.2. Proyectos

Para efectuar una estimación con esta técnica necesitamos un proyecto que estimar, y y un proyecto con valor conocido en la métrica estimada. Así, para estimar el esfuerzo del proyecto A, debemos saber el esfuerzo requerido en el proyecto B. Llamemos este proyecto estimado el proyecto estimado, y al proyecto requerido el proyecto base.

En cada métrica utilizada para generar la estimación, requerimos de a lo menos estos dos proyectos, el proyecto estimado y el proyecto base. Además de estos dos proyectos, podemos tener más proyectos comparados por métricas.

3.3. La Jerarquía

Mientras las métricas comparan proyectos, la jerarquía compara métricas. La estructura de la jerarquía puede tomar varias formas, una jerarquía plana de un solo nivel, con todas las métricas como hijas de la raíz de la jerarquía, o una jerarquía profunda con nodos intermedios que en su ultimo nivel tiene a las métricas.

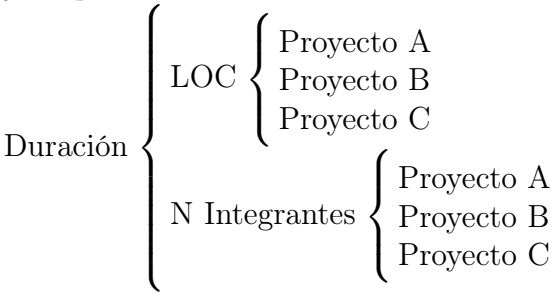
La jerarquía es equivalente a un árbol de nodos. El nodo raíz representa la métrica estimada, y los nodos hojas representan a las métricas. Los demás nodos se encargan de llevar las comparaciones desde los nodos hoja hasta el nodo raíz. Llamamos a los nodos hoja, que representan métricas, nodos métricos. Llamamos a los demás nodos, nodos jerárquicos.

Cada nodo compara. Los nodos jerárquicos comparan otros nodos y los nodos métricos comparan proyectos.

3.4. Ejemplo

Tenemos una jerarquía con dos nodos métricos. Esta jerarquía la usamos para estimar una tercera métrica, y comparamos tres proyectos. Los nodos métricos son las Lineas de Código, o LOC, y el Número de integrantes del equipo. La estimación que queremos es una estimación de Duración en horas y los proyectos de los cuales tenemos información son los proyectos A, B y C.

La jerarquía entonces se conforma del nodo de estimación como raíz, de los nodos métricos como hijos de la raíz, y de los proyectos comparados por métrica como hojas del árbol jerárquico.



Para obtener la estimación de la jerarquía, debemos resolver los nodos métricos, y luego propagar sus resultados hacia la raíz de la jerarquía. La solución de un nodo métrico, es el eigenvector principal de su matriz de comparaciones. Y en el caso de los nodos jerárquicos, la solución consiste en la combinación de las soluciones de sus nodos hijos, ponderadas por el peso de cada hijo. con el peso de un hijo siendo su valor asociado en el eigenvector principal de la matriz de comparaciones de este nodo jerárquico.

Llamamos al eigenvector principal de la matriz de un nodo, la solución de la matriz del nodo. Y llamamos solución del nodo al resultado de la computación del nodo, resultado equivalente a la solución de la matriz en nodos métricos, y resultado que requiere la ponderación de los resultados de los nodos hijos por la solución de la matriz en los nodos jerárquicos. Siguiendo con el ejemplo, supongamos que la solución de la matriz del nodo LOC es [1/6, 2/6, 3/6], pesos asociados a los Proyectos A, B y C respectivamente. Del mismo modo la solución de la matriz del nodo N Integrantes es [1/3, 1/3, 1/3], asociados a los Proyectos A, B, C. Y por último la solución de la matriz del nodo de estimación es [1/6, 5/6], pesos asociados al nodo LOC y al nodo N Integrantes respectivamente. Entonces, la solución de los nodos en es [1/6, 2/6, 3/6] y [1/3, 1/3, 1/3] para los nodos LOC y N Integrantes al ser nodos métricos. Y 1/6*[1/6,2/6,3/6] + 5/6*[1/3,1/3,1/3] al ser un nodo jerárquico, que resulta en [11/36, 12/36, 13/36].

Una vez resuelto el nodo raíz, llamada también la solución de la jerarquía , podemos utilizar esta solución de la jerarquía para computar estimaciones. Este último paso necesita un valor real de un proyecto en esta métrica estimada. Siguiendo con el ejemplo, digamos que estamos estimando el proyecto C, y que sabemos que la duración del proyecto A es 130 horas. El último paso es una regla de tres entre los pesos de la solución de la jerarquía y este

valor real, para conseguir la estimación del valor real. La estimación entonces es = Duración A * peso C / peso A = 130 * 13/36 / 11/36 = 153.63. La estimación de duración para el proyecto C es entonces 153.63 horas.

3.5. Comparaciones

Para conseguir la solución de la matriz de un nodo, necesitamos su matriz, y para tener su matriz necesitamos las comparaciones. Comparaciones entre nodos si la matriz es de un nodo jerárquico, o comparaciones entre proyectos si la matriz es de un nodo métrico. Sea cual sea el tipo de nodo de la matriz, las comparaciones tienen el mismo significado, y la computación de la solución de la matriz es la misma. Computamos su eigenvector principal y luego lo normalizamos.

Supongamos un nodo A con N nodos hijos, los nodos 1, 2, ..., n. Las comparaciones de estos nodos resultan en la matriz de comparaciones

$$M = \begin{bmatrix} w_1/w_1 & \cdots & w_1/w_n \\ \vdots & & \vdots \\ w_n/w_1 & \cdots & w_n/w_n \end{bmatrix}$$

Con cada elemento (i, j) de la matriz correspondiendo a la comparación entre el nodo hijo i con el nodo hijo j, w_i/w_j, la relación entre sus pesos, que son los valores asociados del eigenvector principal asociado a la matriz.

A partir de esta matriz podemos recuperar los pesos de cada nodo al resolver el eigenvector principal de $Aw = nw$, el cual no tiene una solución no trivial ya que n es el valor más grande de A. (Como la matriz A tiene rango unitario, por lo que todos menos uno de los eigenvalues $\lambda_1, \cdots, \lambda_n$ son cero, ya que $\sum_{i=1}^{n} \lambda_i$ =trace$(A) = n$, n es el eigenvalue máximo)

Siguiendo con el ejemplo de la sección anterior, digamos que las comparaciones de los proyectos A, B y C son: A es dos veces el tamaño de B y B es tres veces el tamaño de C. Llamando a la comparación entre A y B comp(A,B), entonces tenemos que comp(A,B) = 2 y comp(B,C) = 3 lo que nos da la matriz

$$\begin{bmatrix} 1 & 2 & x \\ x & 1 & 3 \\ x & x & 1 \end{bmatrix}$$

donde toda la diagonal presenta unos, ya que cada elemento es igual a si mismo. Los elementos marcados con una x los podemos computar a partir de los datos que ya conocemos, suponiendo que la relación de tamaño entre los proyectos es transitiva, entonces comp(A,B) * comp(B,C) = w_A/w_B * w_B/w_C = w_A/w_C = comp(A,C), y ya que comp(X,Y) = w_X/w_Y = 1 / w_Y/w_X = 1 / comp(Y,X), podemos remplazar las x en la matriz por los valores correspondientes.

$$\begin{bmatrix} 1 & 2 & 6 \\ 1/2 & 1 & 3 \\ 1/6 & 1/3 & 1 \end{bmatrix}$$

Matriz con eigenvector principal [6, 3, 1] (de eigenvalor 3). y como la solución de la matriz es el eigenvector principal normalizado, la solución es [6/10, 3/10, 1/10].

3.6. Relación inversa entre una métrica de la jerarquía con la métrica estimada

Hasta ahora hemos tratado la relación entre una métrica utilizada para generar la estimación y la métrica estimada como una relación directa. si la métrica jerárquica es mayor, implica un valor mayor de la métrica estimada. Esta relación entre métricas puede ser también inversa: Si la métrica jerárquica es mayor, la métrica estimada toma un menor valor.

Esto afecta a la computación de la estimación. En caso de ser una relación inversa, invertimos el orden de las comparaciones. Así el elemento en (i,j) pasa a (j,i) y el elemento (j,i) pasa a (i,j).

3.7. Escala de valores para las comparaciones

Las comparaciones suelen provenir de una escala de valores predefinida con una diferencia de no más de un nivel de magnitud. Se comparan ambos nodos con una escala de 1 a 9. El motivo del uso de esta escala es que la capacidad humana de diferenciar con claridad entre nodos degrada cuando la diferencia entre los nodos es mayor a un nivel de magnitud.

En caso de que la escala no baste por una diferencia de magnitud superior a la esperada. Se agrupan los nodos similares (y pequeños) en clusters. estos clusters luego son usados como un todo para compararse con los demás nodos, y los nodos dentro del cluster se comparan entre ellos usando la misma escala, y luego el peso del cluster es propagado al peso de cada nodo del cluster.

$$\text{LOC} \begin{cases} \text{Cluster 1} \begin{cases} \text{Proyecto 1} \\ \text{Proyecto 2} \end{cases} \\ \text{Proyecto 3} \end{cases}$$

4. AHP simplificado: modificaciones a la técnica

A continuación se comentan las modificaciones a la técnica AHP que se propone en esta memoria. Estas modificaciones tienen los siguientes objetivos.

1. Simplificar el algoritmo utilizado para calcular los pesos de los nodos hijos. Esto nos permitirá disminuir el número mínimo de comparaciones por proyecto nuevo a 1.
2. Aprovechar la persistencia de las comparaciones de una estimación a otra, basándonos en el hecho de que la comparación entre dos proyectos terminados se mantiene constante a lo largo del tiempo.
3. Aprovechar la persistencia de las jerarquías de una estimación a otra, basándonos en el hecho de que la jerarquía para el cómputo de una estimación depende de las métricas y no de los proyectos que compara.
4. Creación de una jerarquía default para caracterizar los proyectos y generar estimaciones.

Estas modificaciones apuntan a disminuir el esfuerzo requerido para completar una estimación por parte del usuario. En el caso original, la se tiene que armar la jerarquía y hacer la comparación entre todos los proyectos para todas las métricas cada vez que se quiera hacer una estimación. En el caso modificado, suponiendo que se han hecho estimaciones anteriores y ya se han hecho jerarquías y comparaciones, solo se tiene que comparar el proyecto siendo estimado con un solo proyecto extra para cada métrica, cada vez que se quiera hacer una estimación.

Con la simplificación del algoritmo, lo que buscamos es disminuir la cantidad de comparaciones necesarias para resolver una matriz. Actualmente sabemos que la diagonal de la matriz de comparaciones es siempre una diagonal de unos. Sabemos también que la mitad inferior izquierda de la matriz es el inverso de la mitad superior derecha de la matriz (ya que el elemento (i,j) es el inverso de (j,i), para todo i,j índices de la matriz).

Veremos ahora que la cantidad de comparaciones por métrica necesarias para resolver una matriz es a lo menos una, la comparación entre el proyecto base y el proyecto estimado. Veremos primero la idea de clausura, que sirve de explicación para estas simplificaciones.

4.1. Clausura de comparaciones

Denotando una comparación como un par (x, y), con $(x, y) = w_x/w_y$

Par transitivo

Un par transitivo es un par de comparaciones capaz de generar una nueva comparación. teniendo las comparaciones (a,b) y (b,c), por transitividad tenemos que $(a, b) * (b, c) = (w_a/w_b) * (w_b/w_c) = w_a/w_c = (a, c)$.

Par pivote

Del mismo modo, el par pivote también es un par de comparaciones que genera una nueva comparación. teniendo las comparaciones (a,b) y (a,c), por transitividad tenemos que $(a, c)/(a, b) = (w_a/w_c)/(w_a/w_b) = w_b/w_c = (b, c)$

El par pivote y el par transitivo son equivalentes. Si tenemos el par transitivo (a,b) y (b,c), también tenemos el par pivote (b,a) y (b,c). Consideramos estos tipos de pares de comparaciones diferentes porque en general nos interesa mantener un orden entre los elementos comparados, orden utilizado constantemente en los algoritmos de la implementación.

Clausura

Una clausura de una matriz de comparaciones es un conjunto de elementos que pueden ser comparados entre ellos en la matriz, utilizando las comparaciones actuales (elementos que son nodos si son comparados en un nodo jerárquico, o elementos que son proyectos si son comparados en un nodo métrico).

Si tenemos una matriz con los elementos: a,b,c,d,e y con las comparaciones (a, b), (a, c) y (d, e), tenemos las clausuras (a, b, c) y (d,e) ya que a puede ser comparado con b y con c, pero no con d ni con e. Y d se puede comparar únicamente con e. La idea de una clausura nos permite saber si podemos comparar dos elementos, dado un conjunto de comparaciones.

En el ejemplo anterior no tenemos la comparación (b,c) pero sabemos que podemos comparar ambos elementos ya que pertenecen a la misma clausura, y en efecto, como tenemos el par pivote (a,b) y (a,c), podemos computar la comparación (b,c), (a,c)/(a,b) = (b,c)

Un elemento pertenece a una sola clausura y no a más de una. En caso de que pertenezca a dos, podemos utilizar comparaciones con este mismo elemento como pares transitivos o pivotes para generar todas las demás comparaciones entre los elementos de una clausura con la otra, terminando con una sola clausura con los elementos de ambas, mostrando la imposibilidad de que un elemento pertenezca a más de una clausura.

Generación de comparaciones

Una clausura de comparaciones es capaz de generar una matriz de comparaciones completa. La matriz de comparaciones tiene tres partes. La diagonal, que compara cada elemento con si mismo. La mitad inferior izquierda, y la mitad superior derecha. La mitad inferior izquierda es la inversión de la mitad superior derecha. para cada elemento (i,j) de la matriz, con i diferente a j, tenemos el elemento (j,i), donde M(i,j) = 1 / M(j,i). En el caso particular de la diagonal, cada comparación tiene un valor de 1.

De modo que para definir la matriz completa, nos basta definir la mitad superior derecha (o equivalentemente, la mitad inferior izquierda).

Supongamos ahora que la matriz esta conformada por los elementos del conjunto A, con $M_{ij} = w_{A_i}/w_{A_j}$ (el elemento i,j de la matriz compara el elemento i con el elemento j del conjunto A). Y que el conjunto A forma una clausura. Entonces, debido a la definición de la clausura, para cualquier i,j somos capaces de generar cualquier comparación (i,j) de la matriz. Otra forma de verlo es que una clausura genera una matriz de comparaciones entre sus elementos.

Clausura en nodos jerárquicos

El objetivo de los nodos jerárquicos es la agregación de las soluciones de los nodos hijos. En este contexto podemos ver una clausura entre todos sus hijos como necesaria, definiendo la relación entre cada una de las diferentes soluciones de sus hijos.

4.2. Comparaciones relajadas

Lo que buscamos es saber el número mínimo de comparaciones necesarias requeridas para resolver una matriz de comparaciones.

Requiriendo una sola comparación para cada proyecto nuevo podemos calcular una estimación para el proyecto nuevo. Esta única comparación es suficiente y puede hacerse con cualquier proyecto anterior ya que todos son puntos de referencia. Como esta comparación genera una estimación del proyecto nuevo, este nuevo proyecto también se vuelve un punto de referencia.

4.3. Uso de valores reales

En el caso de los nodos métricos, podemos mantener el uso de la escala de valores para hacer comparaciones entre proyectos. El uso de la escala ayuda en caso de que se estime la relación entre tamaños porque si tenemos los valores reales de los proyectos para la métrica, es mejor, al dar un resultado más preciso, utilizar estos valores reales para calcular las comparaciones.

La comparación, en caso de usarse valores reales, se convierte en la relación entre estos valores reales de los proyectos comparados. Si el proyecto A tiene un equipo de 20 personas y el proyecto B tiene un equipo de 5 personas, entonces la comparación para esta métrica de estos proyectos es comp(A,B) = 4.

4.4. Reutilización de información

La generación de una estimación, y la generación de la siguiente estimación pueden tener muchas partes en común. Partes en común que pueden ser reutilizadas entre estimación y estimación. Es más fácil ver estas partes si dividimos el proceso de generar una estimación en etapas.

- elegir un proyecto a estimar
- elegir una métrica a estimar
- elegir un proyecto base
- armar la jerarquía
- hacer las comparaciones entre proyectos por métrica en la jerarquía, y si se tienen valores reales, computar las comparaciones necesarias.
- asegurarse de que tanto el proyecto base como el proyecto a estimar estén en las mismas clausuras en cada métrica de la jerarquía
- hacer las comparaciones entre nodos en la jerarquía
- resolver la jerarquía

- relacionar el proyecto base, y la solución de la jerarquía para conseguir la estimación del proyecto a estimar.

La jerarquía utilizada para generar la estimación, así como sus comparaciones, son independientes de la estimación que estemos haciendo. Esto nos permite reutilizar una misma jerarquía en diferentes oportunidades, para diferentes estimaciones. Estimaciones que mantengan constante la misma métrica estimada, ya que una jerarquía de estimación solo tiene sentido en el contexto de estimación de la métrica para la que fue creada.

De modo que si queremos hacer estimaciones de diferentes métricas, requeriremos diferentes jerarquías, lo que nos lleva al siguiente componente que podemos reutilizar, las comparaciones entre proyectos en una métrica. Una jerarquía que estima el tiempo puede requerir comparar proyectos por el tamaño de sus equipos del mismo modo que una jerarquía que estima el esfuerzo requiere estas comparaciones. Reutilizamos en consecuencia las comparaciones entre proyectos entre diferentes jerarquías.

La resolución de la jerarquía, por otro lado, depende del proyecto que estemos estimando y del proyecto base que usemos. No podemos mantener el proyecto base constante, ya que no podemos asegurar que el proyecto que estimemos esté en la misma clausura que el proyecto base, requisito para poder resolver la jerarquía.

5. Implementación de la técnica AHP Simplificada

Esta sección se centra en la implementación algorítmica de las diferentes partes de la técnica AHP simplificada. Esta es la implementación utilizada por la herramienta, la cual se muestra en capítulos siguientes.

5.1. Jerarquías y métricas

Como las métricas son reutilizadas por las jerarquías, decidimos mantenerlas separadas, haciendo que las jerarquías apunten a las métricas que usa.

5.2. Tipos de nodo

Los tipos de nodos implementados reflejan los tipos de nodos descritos en la sección anterior, tenemos los nodos jerárquicos que comparan nodos, y tenemos los nodos métricos que comparan proyectos.

La principal utilidad esperada de los nodos jerárquicos es la creación de clusters para facilitar la comparación entre métricas, agrupándolas por tamaño (como se explica en [17]).

Por el lado computacional, la separación facilita la implementación, ya que los nodos jerárquicos cumplen un rol de agregación de los nodos que tengan como hijos, mientras que los nodos métricos son los nodos bases que manejan la base de las computaciones. Por ejemplo, la carga de métricas en la jerarquía parte al nivel del nodo raíz, que pasa el trabajo de cargar las métricas a sus hijos, al igual que todos los nodos jerárquicos. Por otro lado los nodos métricos buscan su métrica correspondiente para cargarla y poder hacer la estimación.

La clausura de proyectos y la inversión de matrices son más ejemplos de características que solo tienen sentido en un tipo de nodo (nodos métricos), y no en el otro.

En su estado actual, la implementación consiste en una interfaz común para ambos tipos de nodos, mientras que cada nodo implementa los métodos de la interfaz de forma diferente. En un principio ambos nodos eran el mismo, y se diferenciaban según si tenían hijos (nodo jerárquico) o no (nodo métrico). Luego se optó por la idea de la herencia, teniendo al nodo métrico heredando funcionalidad del nodo jerárquico. Por último, por una decisión de estilo centrado en la composición de clases por sobre la herencia derivó en el uso de una interfaz común. Al final resulto ser que no se ganaba nada con la herencia, por lo que el cambio resultó en una mayor claridad en el funcionamiento de cada nodo.

5.3. Comparaciones

Como las comparaciones funcionan del mismo modo para ambos tipos de nodos, estos tienen una implementación separada con su propia interfaz, utilizada por los nodos.

La estructura utilizada es un hash de hashes (o equivalentemente, un hash de dos llaves), con la primera llave respondiendo al identificador del elemento izquierdo de la comparación, y la segunda llave respondiendo al identificador del elemento derecho de la comparación. Al tener las dos llaves, accedemos al valor de la comparación.

El motivo de esta estructura es para disminuir la cantidad de información guardada. En vez de guardar toda la matriz de comparaciones, guardamos únicamente las comparaciones de las que tenemos información. La diagonal siempre es una diagonal de unos, y la mitad inferior izquierda es la mitad superior derecha pero inversa. Por lo que nos basta guardar solo una mitad (superior o inferior), y de esta mitad, no necesitamos guardar los valores default, solo los valores que no son los default. Para lograr esto, guardamos el hash de hashes como ya explicado, y además guardamos el orden de los elementos comparables, asegurándonos así de guardar siempre la versión de la comparación de la mitad superior (o inferior, lo importante es mantenerse constante en la mitad que guardamos).

Desde un punto de vista de transferencia de datos, mover una menor cantidad de datos es mejor. Además, como no se espera que esta técnica sea utilizada con cantidades grandes de información (como se explica en la discusión), el coste extra en tiempo de computación que lleva la computación de la matriz completa es insignificante.

También, para mantener la consistencia visual entre sesión y sesión del usuario, y entre diferentes estimaciones, guardamos también el orden de los elementos comparables.

5.4. Computación de una clausura

Como se explica en el apartado de clausuras, las clausuras nos indican la validez de un grupo de comparaciones para la generación de una estimación. Si no tenemos las comparaciones suficientes, entonces podemos avisarle al usuario qué métrica requiere más comparaciones, y en caso de tener suficientes comparaciones, sabemos qué proyectos incluir en la matriz para generar en última instancia la estimación.

En esta implementación, las clausuras son utilizadas únicamente en los nodos métricos y se permite que en los nodos jerárquicos no se tengan comparaciones, resultando en que cada nodo hijo sea parte de su propia clausura. En la sección anterior comentamos la necesidad de la clausura en los nodos jerárquicos para siempre tener una relación entre sus nodos hijos. El algoritmo utilizado para la aproximación del eigenvector principal, sumado a la normalización que recibe este resultado entrega una relación base entre los elementos de un nodo, tenga o no tenga comparaciones. Esta relación entre nodos se acomoda a cualquier número de clausuras presente en las comparaciones, ya sea una para cada nodo hijo, o una clausura con todos los hijos. El propósito de la clausura en un nodo jerárquico es tener una relación entre los nodos hijos, y gracias al algoritmo utilizado siempre tenemos esta relación entre los hijos de un nodo jerárquico, haciendo innecesario el tener una sola clausura.

Donde son necesarias las clausuras es en los nodos métricos. Para generar la estimación final con la jerarquía, necesitamos que en cada métrica utilizada por la jerarquía, tanto el proyecto base como el proyecto estimado pertenezcan a la misma clausura.. Para computar esta clausura, lo que hacemos es computar todas las clausuras que tiene un nodo, y luego buscamos la que tenga ambos proyectos en ella. Si no está le avisamos al usuario para que una los proyectos en una clausura.

El algoritmo para computar las clausuras es el siguiente.

1. construimos un set para cada elemento a del conjunto de comparaciones. El set lo

componen los elementos en las comparaciones en que está presente este elemento como primer elemento: comparaciones del tipo (a,x), para el elemento a, y para cualquier otro x. En otras palabras, armamos los sets utilizando cada elemento como pivote.

2. luego iteramos sobre los elementos de todas las comparaciones.
3. para cada elemento, combinamos todos los sets que contengan el elemento.
4. cuando no quedan más elementos por iterar, terminamos.

Como resultado, cada set que queda es una clausura.

El algoritmo hace uso del principio de las clausuras que asegura de que un elemento solo puede pertenecer a una sola clausura. Como cada set siempre contiene elementos que se conectan por comparaciones, entonces, si un elemento aparece en más de un set, la única posibilidad es que ambos set sean parte de la misma clausura. Una vez que aseguramos este principio para cada elemento, podemos asegurar de que cada set restante es una clausura.

5.5. Generación de la matriz de comparaciones

Una vez que tenemos la validación de un conjunto de comparaciones, validación que nos dice que somos capaces de generar una matriz completa a partir de las comparaciones que tenemos, procedemos a generar la matriz.

El algoritmo para generar la matriz es el siguiente.

1. armamos una lista de las comparaciones actuales, la lista A.
2. probamos todos los pares de comparaciones de esta lista y armamos una nueva lista, la lista B, con las comparaciones generadas y las comparaciones de la lista A.
3. si la lista B es igual a la lista A entonces no generamos ninguna nueva comparación, por lo que terminamos.
4. si las listas son distintas, entonces volvemos al paso 2, pero ahora la nueva lista A es la lista B.

5.6. Computación aproximada de los eigenvectores de la matriz

La solución de la matriz se implementa con una resolución aproximada del eigenvector principal.

El algoritmo consiste en computar la suma de los elementos por columna. Luego normalizar las columnas por este valor. Y por último promediar por fila, para terminar con el peso del elemento i como el resultado de la fila i de la matriz.

5.7. Propagación de resultados avanzando en la jerarquía

Cuando se comparan nodos, lo que se hace es reajustar los valores recibidos desde los nodos inferiores. Supongamos que tenemos una jerarquía de nodos X,Y,Z, con Y y Z hijos de X, supongamos también que Y y Z son nodos métricos que comparan los proyectos A,B y C, y que la solución a sus matrices es:

para Y

$$Y_A = 0{,}1; Y_B = 0{,}4; Y_C = 0{,}5$$

para Z

$$Z_A = 0{,}2; Z_B = 0{,}2; Z_C = 0{,}6$$

por último, la solución de la matriz de comparaciones de X es

$$X_Y = 0{,}3; X_Z = 0{,}7$$

para propagar los resultados obtenidos de las matrices de Y y Z, ponderamos sus resultados por los pesos en X, y luego sumamos

$$X_A = X_Y * Y_A + X_Z * X_A = 0{,}3 * 0{,}1 + 0{,}7 * 0{,}2 = 0{,}03 + 0{,}14 = 0{,}17$$

$$X_B = X_Y * Y_B + X_Z * X_B = 0{,}3 * 0{,}4 + 0{,}7 * 0{,}2 = 0{,}12 + 0{,}14 = 0{,}26$$

$$X_C = X_Y * Y_C + X_Z * X_C = 0{,}3 * 0{,}5 + 0{,}7 * 0{,}6 = 0{,}15 + 0{,}42 = 0{,}57$$

resultando en

$$X_A = 0{,}17; X_B = 0{,}26; X_C = 0{,}57$$

con lo cual, podemos seguir subiendo en la jerarquía con los nuevos valores de los proyectos.

Cabe señalar que en cada paso, los resultados de los nodos hijos que necesitamos son únicamente de los proyectos que queremos comparar. Si por ejemplo estamos comparando el proyecto A con el proyecto B, entonces podemos deshacernos del peso del proyecto C y solo subir por la jerarquía los pesos de los proyectos relevantes (A y B).

5.8. Requisitos mínimos para la generación de una estimación

Para hacer estimaciones, lo que requerimos es que tanto el proyecto estimado como el proyecto base pertenezcan a las mismas clausuras. Para cada métrica de la jerarquía, ambos proyectos deben pertenecer a la misma clausura. En otras palabras, los proyectos deben ser comparables en cada métrica de la jerarquía.

Sobre las comparaciones de los nodos jerárquicos no hay restricciones que impidan al algoritmo computar una estimación. El algoritmo que computa la aproximación al eigenvector principal tiene la propiedad de entregar un valor para todos los hijos del nodo. En caso de que un hijo de un nodo jerárquico no tenga comparaciones, entonces la solución que viene de él es ponderada por 1/n, con n el número de hijos del nodo.

5.9. Etapas en la generación de una estimación

Viendo ahora las etapas de estimación según los datos que requiere.

Datos requeridos por etapa de la computación

1. elección de proyecto a estimar: Lista de proyectos. El proyecto que estimemos puede ser cualquiera de los que se tenga información.
2. elección de métrica a estimar: Lista de métricas. La técnica permite la estimación de cualquiera de las métricas de la que se tenga información. El requerimiento es que el proyecto base escogido tenga un valor en esta métrica.
3. elección de proyecto base: Lista de proyectos. El proyecto base debe ser uno que tenga un valor real en la métrica escogida. Un requerimiento extra es que tanto el proyecto a estimar como el proyecto base deben ser parte de la misma clausura, para cada métrica utilizada en la jerarquía.
4. elección de jerarquía: Lista de jerarquías. La jerarquía puede ser cualquiera de las guardadas en el sistema. Las jerarquías son construidas con una métrica a estimar en mente. Si bien podemos utilizar cualquier jerarquía para cualquier estimación, utilizar una jerarquía para estimar el costo de un proyecto cuando fue construida para la estimación de tiempo no genera una estimación útil.
5. para la jerarquía: Estructura jerárquica de los nodos, comparaciones de los nodos jerárquicos, y los identificadores de los nodos métricos. En esencia, necesitamos toda la información que es particular a la jerarquía usada para la estimación, así como los identificadores de las métricas que utiliza la jerarquía. Identificadores llevados por los nodos métricos.
6. por cada nodo métrico: Comparaciones de la métrica y valores reales para la métrica de los proyectos. Tanto las comparaciones como los valores reales son los utilizados para conseguir las clausuras de la métrica, así como para conseguir la solución del nodo.

Varias de estas etapas pueden usar menos información, si tomamos en cuenta los requerimientos. Para el proyecto base, solo nos interesan los proyectos que tengan un valor real en la métrica estimada. Para la jerarquía, solo nos interesan las métricas que forman parte de esta jerarquía. Para cada métrica en la jerarquía, solo nos interesan los proyectos que forman parte de la clausura del proyecto base y del proyecto estimado.

La implementación hecha no considera estos requerimientos. Para permitir mutabilidad en las diferentes partes que generan la estimación se cargan todas las métricas, así el usuario puede modificar la jerarquía cambiando las métricas que usa la jerarquía. Para cada métrica, se cargan todos los proyectos, así el usuario puede modificar las clausuras como vea necesario. El lado negativo que conlleva la mutabilidad es la necesidad de cargar una mayor cantidad de información.

Etapa final

Una estimación sobre un proyecto en una métrica (sea tiempo, costo, tamaño, etc) requiere de a lo menos otro proyecto (el proyecto base) con el cual ser comparado, y que este proyecto base tenga un valor real en dicha métrica, para poder aplicar el último paso de la estimación, luego de haber generado los pesos de los proyectos a partir de la jerarquía. El último paso es una simple regla de tres entre los valores del peso del proyecto estimado, el peso del proyecto base, y el valor real del proyecto base en la métrica en la que se está estimando.

6. Herramienta

6.1. Arquitectura

Para una fácil adopción y uso por los usuarios se optó desarrollar la herramienta como una aplicación web.

El lenguaje utilizado para su construcción es Angular, principalmente por su uso de typescript que tiene soporte de tipos, lo que facilita un desarrollo con menos bugs, en parte por su estructura de proyecto que separa la lógica de lo visual en diferentes archivos, y en parte por el nivel de experiencia que poseo con dicho framework.

Para la interfaz gráfica se utiliza angular-material al ser una extensión de angular.

La base de datos se mantiene en Firebase. Esta base de datos es gratuita en niveles bajos de demanda de red, lo que la hace perfecta para desarrollos tempranos de aplicaciones.

El desarrollo de software siguió un desarrollo orientado a la composición de clases, por sobre la herencia. Se mantiene una amplia gama de tests para asegurar el funcionamiento de los componentes, y se sigue el principio de que el código y los tests son la fuente primaria de conocimiento sobre el software. No se usan comentarios, ya que el papel de explicar que es lo que hace cada parte se le deja al código mismo y a los tests.

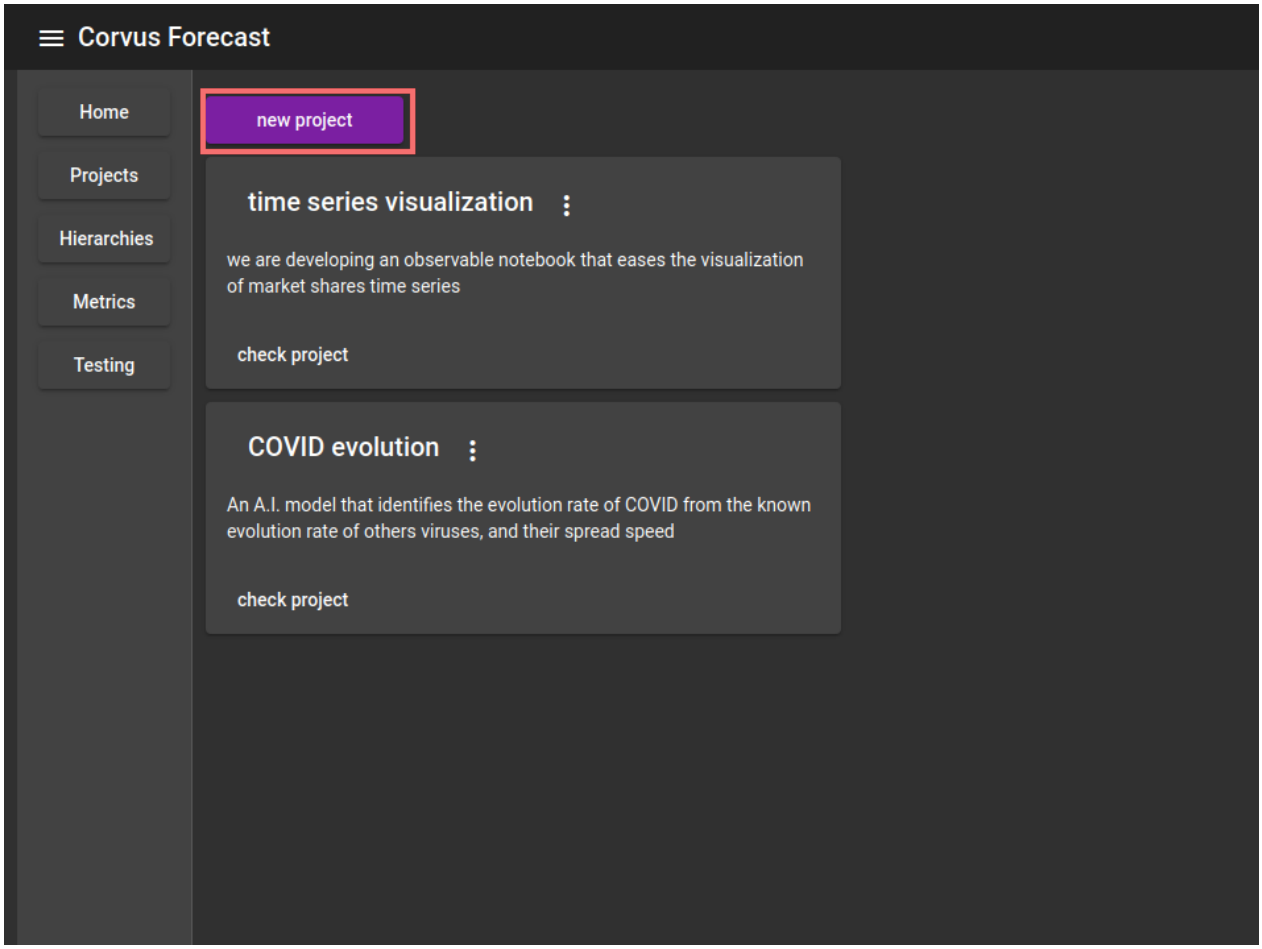
6.2. Casos de uso

Caso de uso 1: Estimación de un proyecto

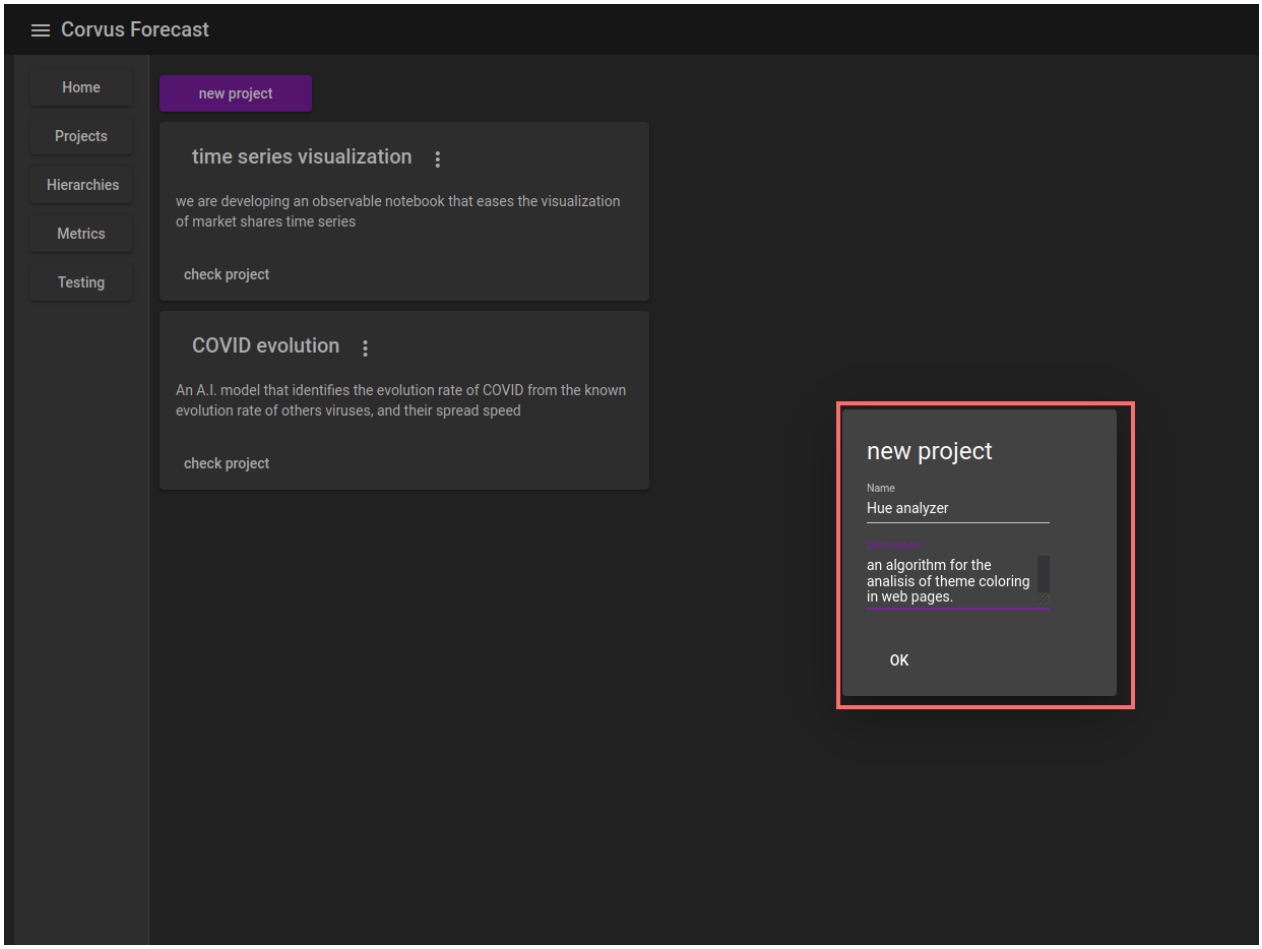
En este ejemplo crearemos una nueva estimación de tiempo para un nuevo proyecto.

El proyecto que queremos estimar es un nuevo algoritmo que nos permite perfilar páginas web según la paleta de colores que utiliza, así como su distribución de colores entre los elementos.

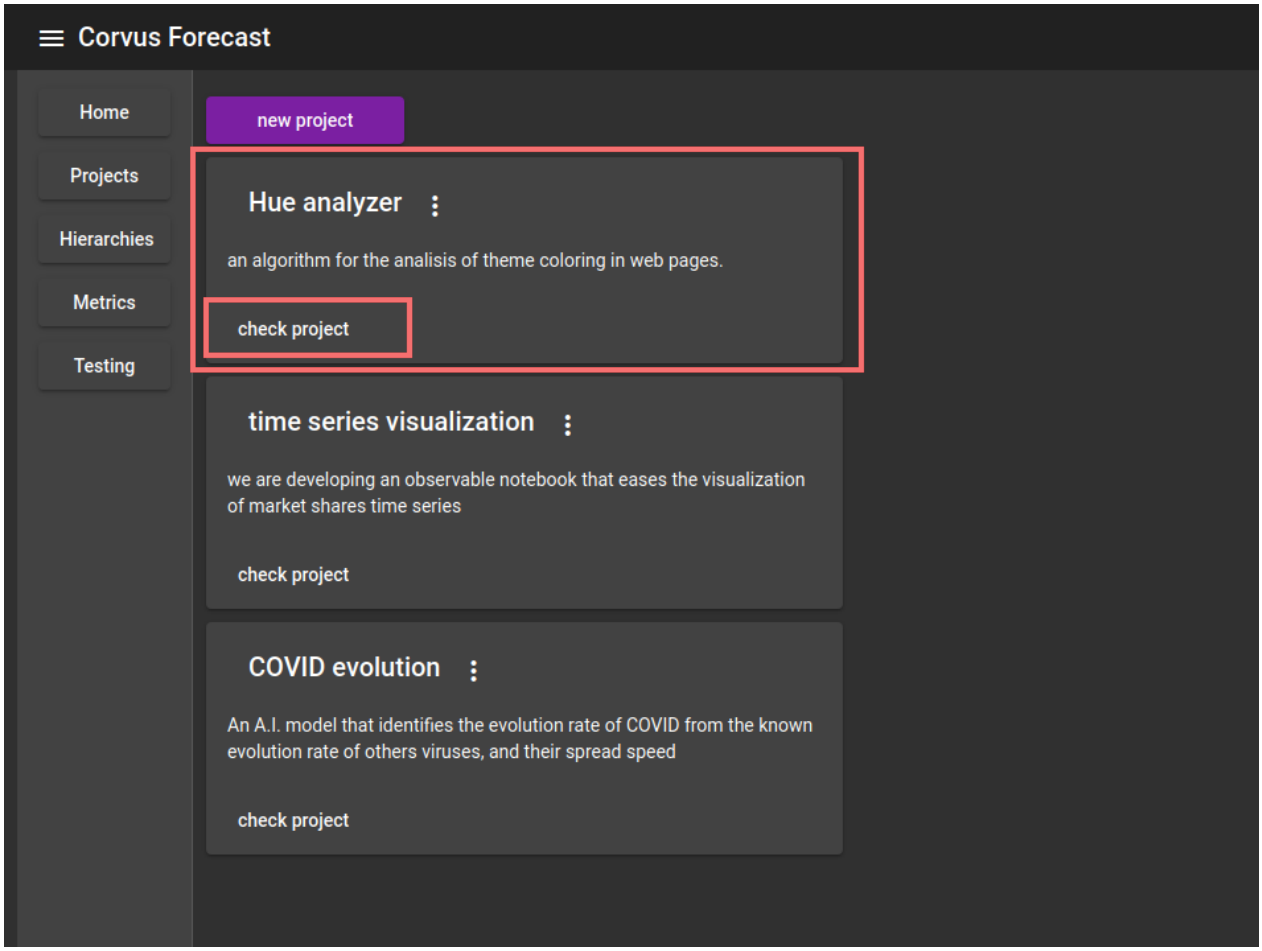
Lo primero que hacemos es crear el nuevo proyecto, en la vista de proyectos



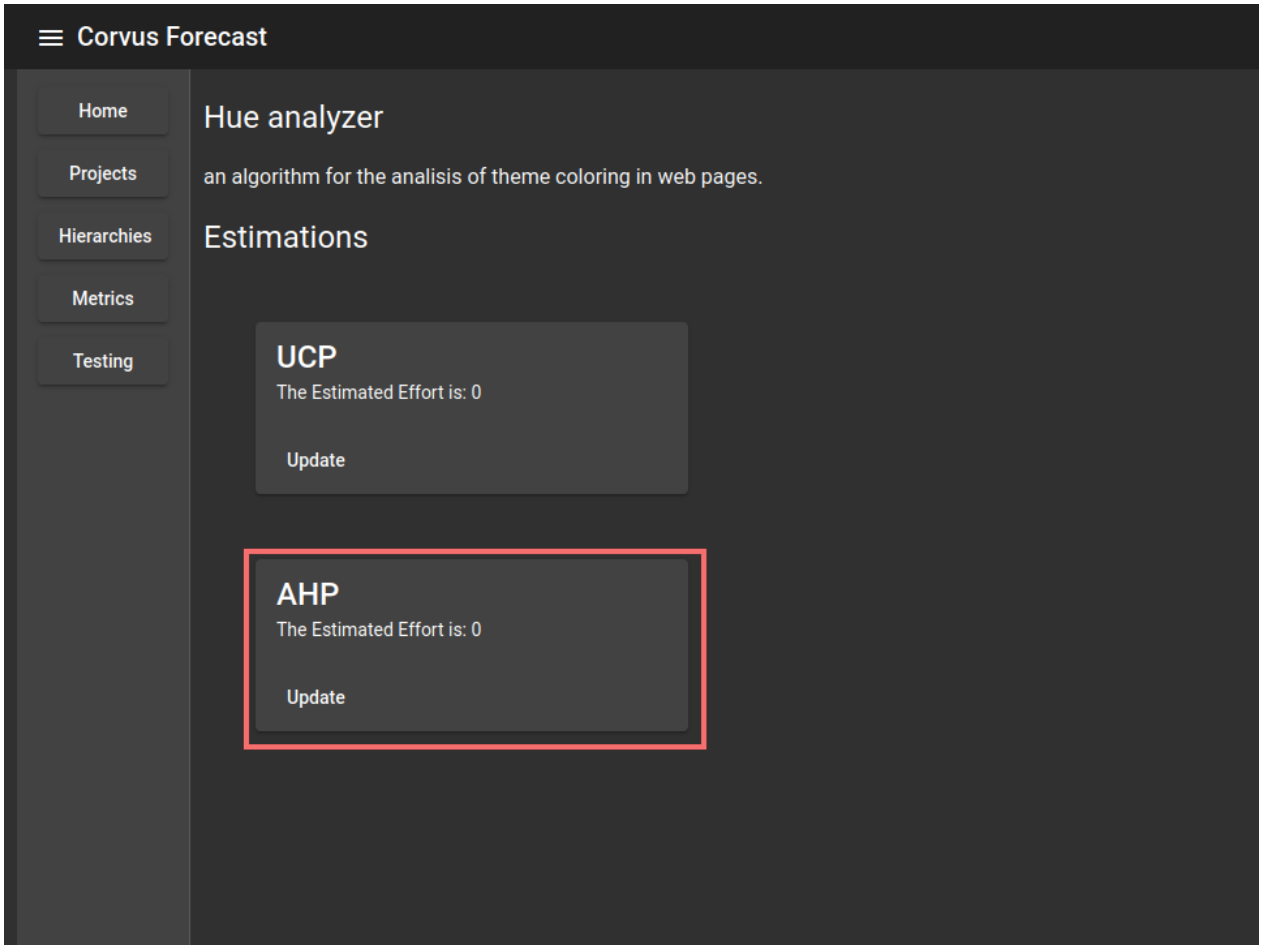
El diálogo para la creación de proyectos nos pide un nombre y una descripción de proyecto. Lo llenamos con la información de este nuevo proyecto.



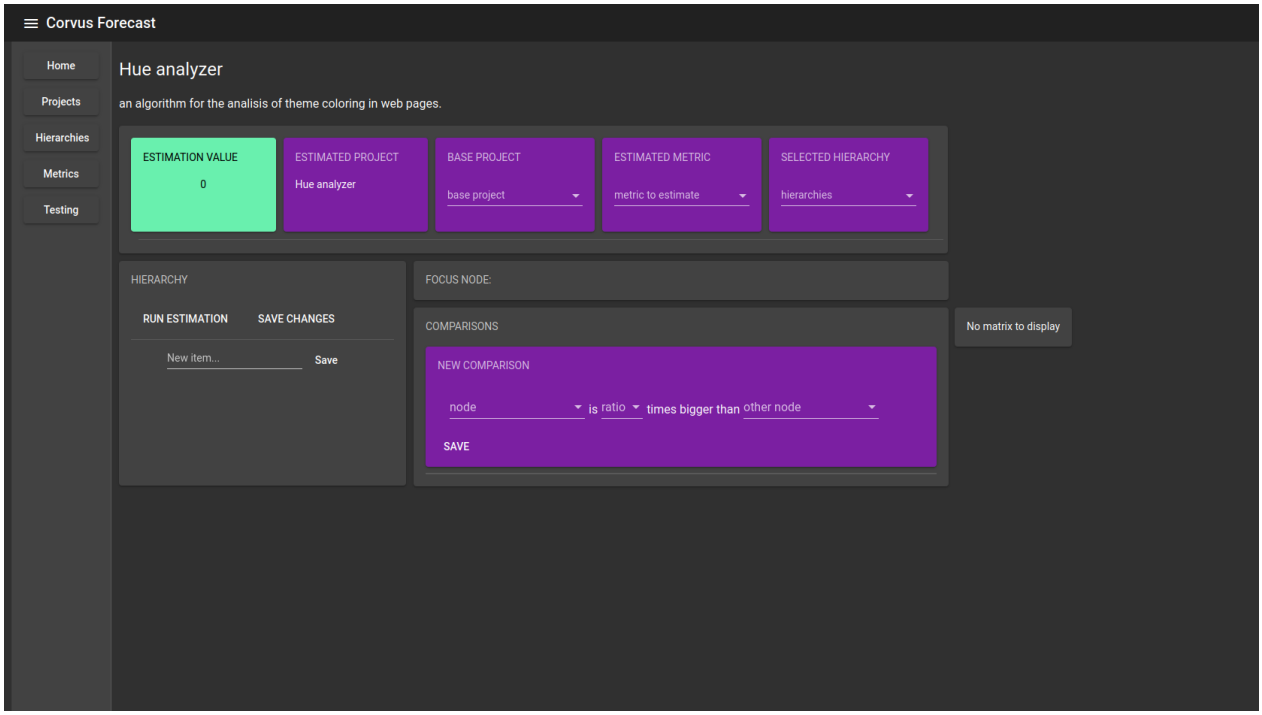
Una vez creado este nuevo proyecto, accedemos a él para hacer la estimación.



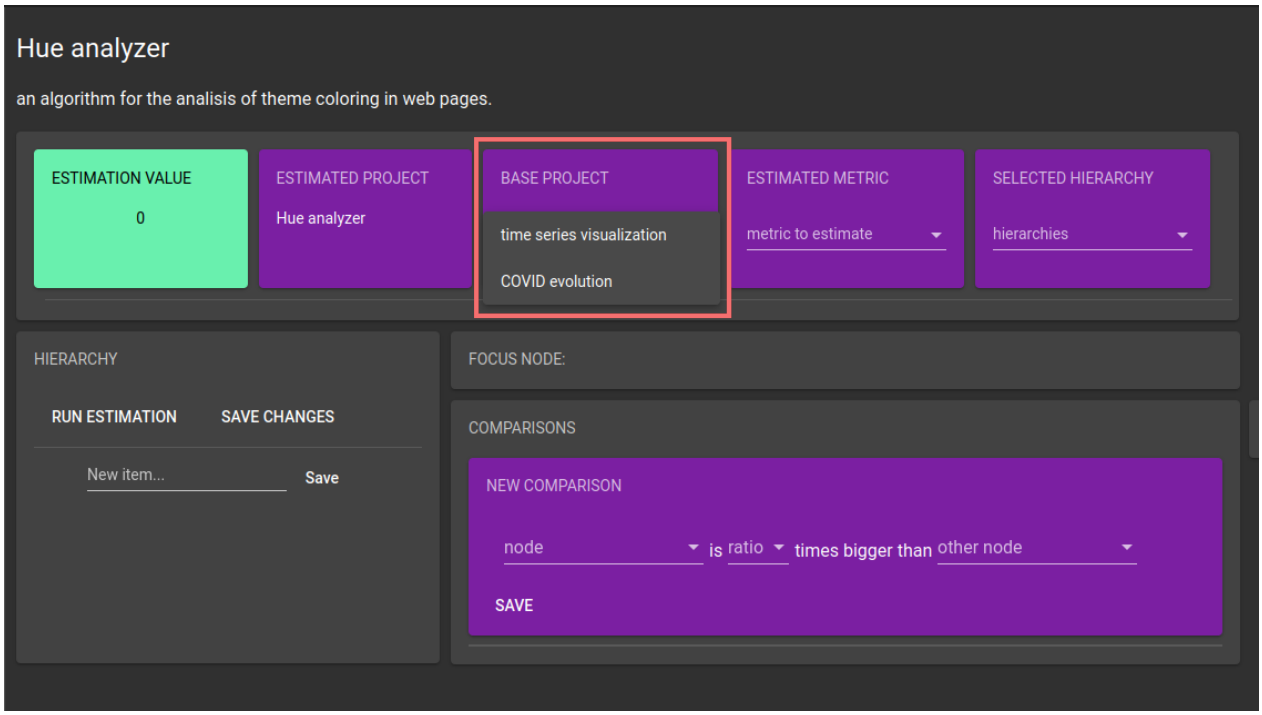
En la página del proyecto, tenemos la opción de hacer la estimación con la técnica UCP o la técnica AHP simplificada, optamos por hacer la estimación con la técnica AHP simplificada.



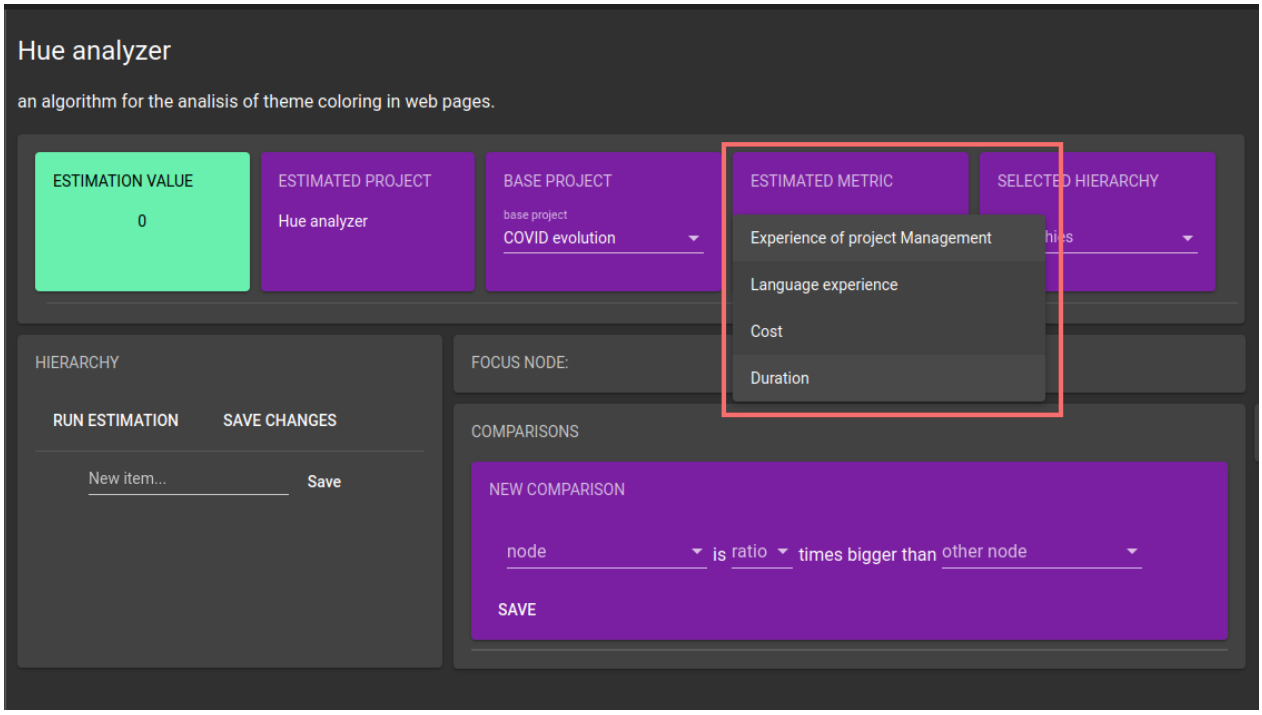
Llegamos a la página para hacer estimaciones. Podemos ver tres secciones principales, la sección superior con un grupo de elecciones, la sección inferior izquierda con la jerarquía, y la sección inferior derecha con las comparaciones. No hemos elegido ninguna jerarquía para la estimación, por lo que no hay cargada ninguna jerarquía ni ninguna comparación.



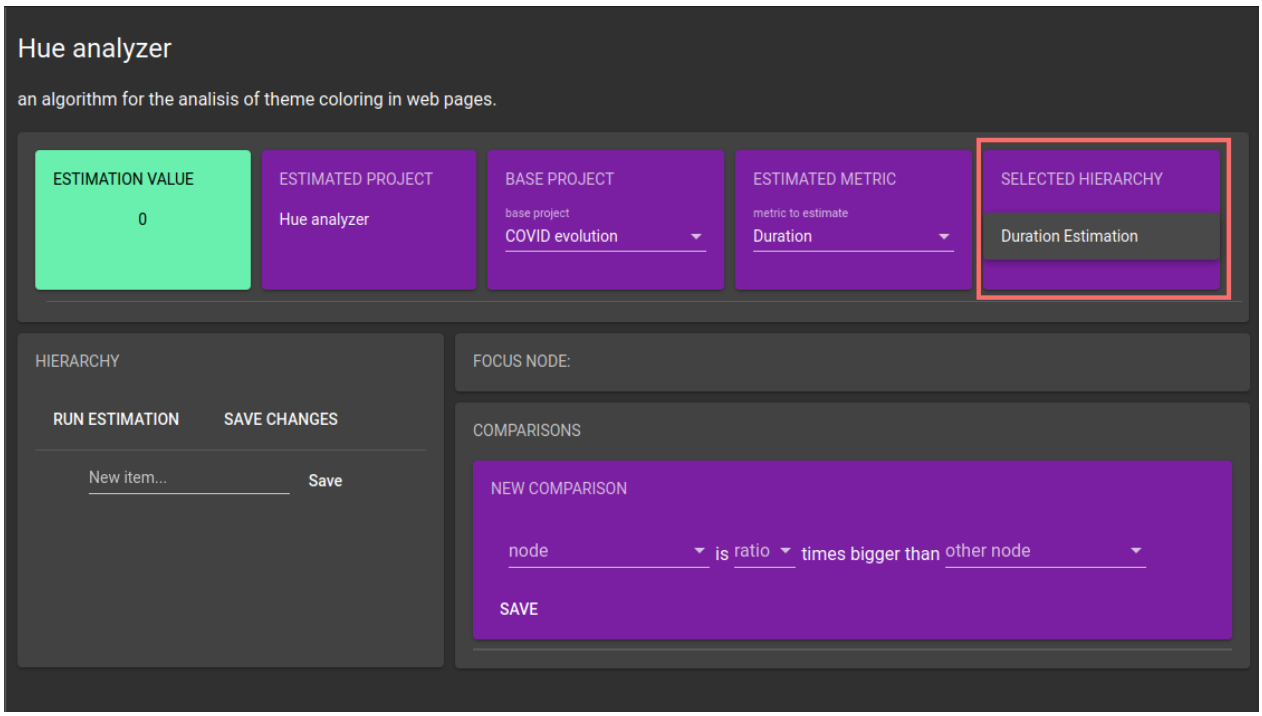
Comenzamos con la estimación, eligiendo el proyecto base sobre el cual basar la estimación. Elegimos el proyecto COVID evolution el cual tiene un valor real de duración, requerimiento para ser proyecto base.



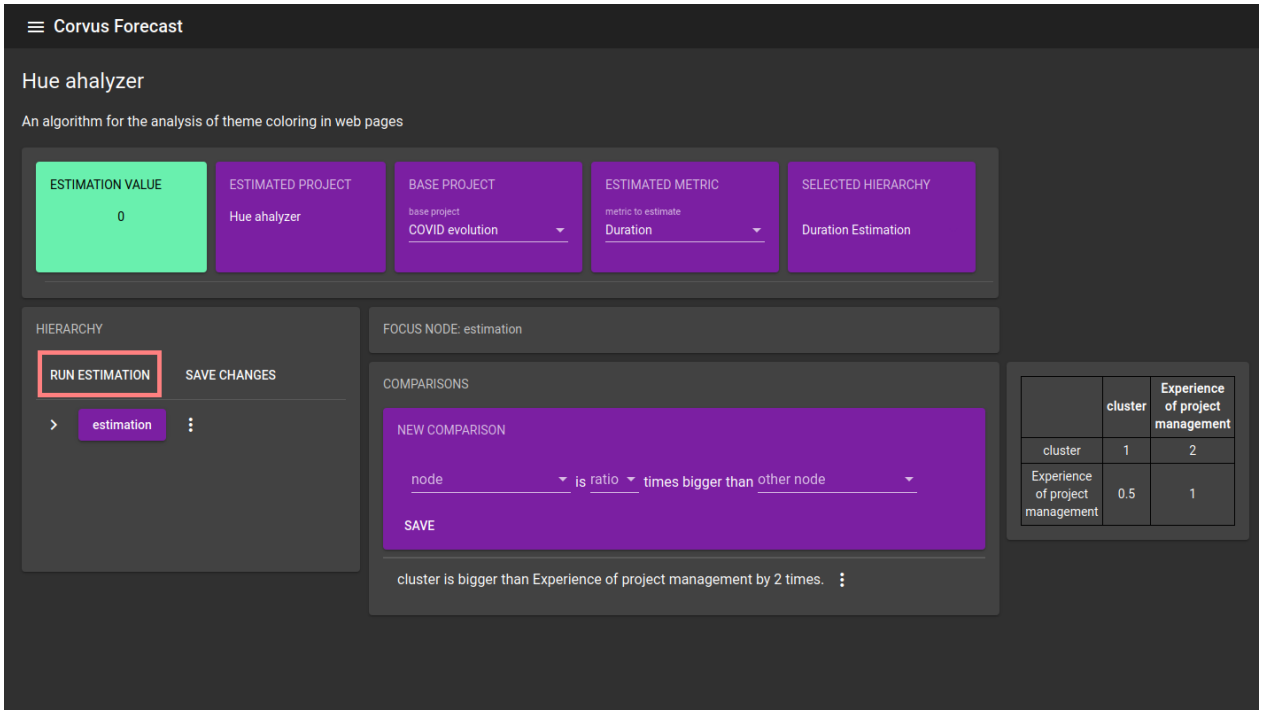
Continuamos con la elección de la métrica que queremos estimar. Queremos estimar la duración del proyecto, así que la escogemos entre las métricas en el sistema.



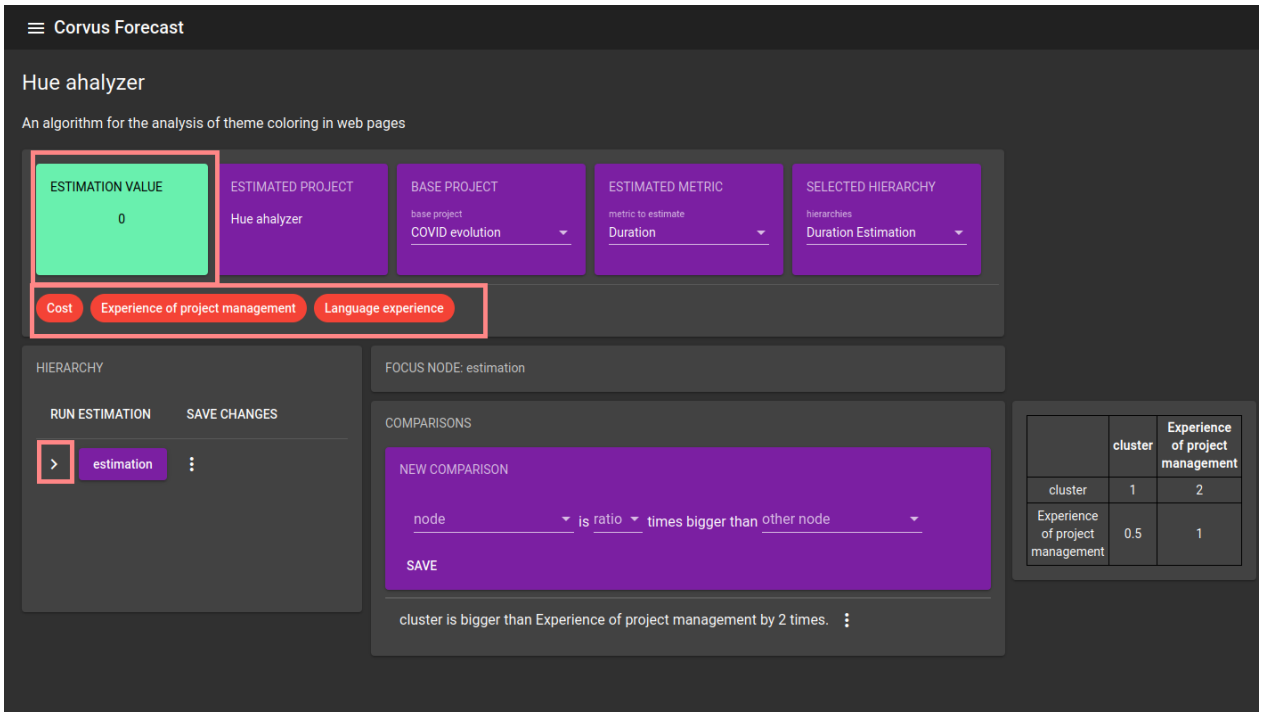
Terminamos las elecciones al elegir la jerarquía que usaremos para hacer la estimación. Tenemos solo una jerarquía disponible en el sistema, así que la usamos.



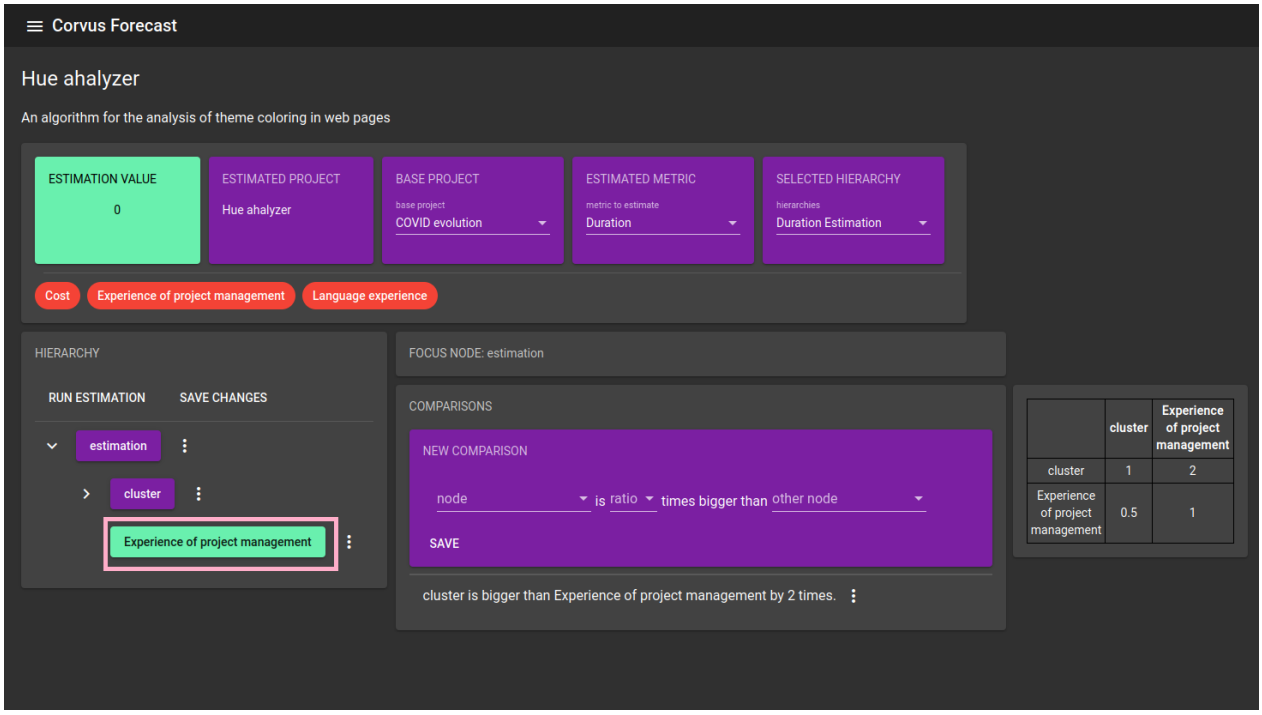
Una vez tomadas todas las decisiones, corremos el algoritmo para obtener la estimación.



La estimación no cambia de 0, y bajo esta podemos ver una lista de problemas. Lo que nos indica el sistema es que para cada una de las métricas en rojo, el proyecto estimado y el proyecto base no pertenecen a la misma clausura. Esto tiene sentido ya que acabamos de crear el proyecto que estamos estimando. Procedemos entonces a expandir la jerarquía para hacer que el proyecto base y el proyecto estimado estén en la misma clausura.

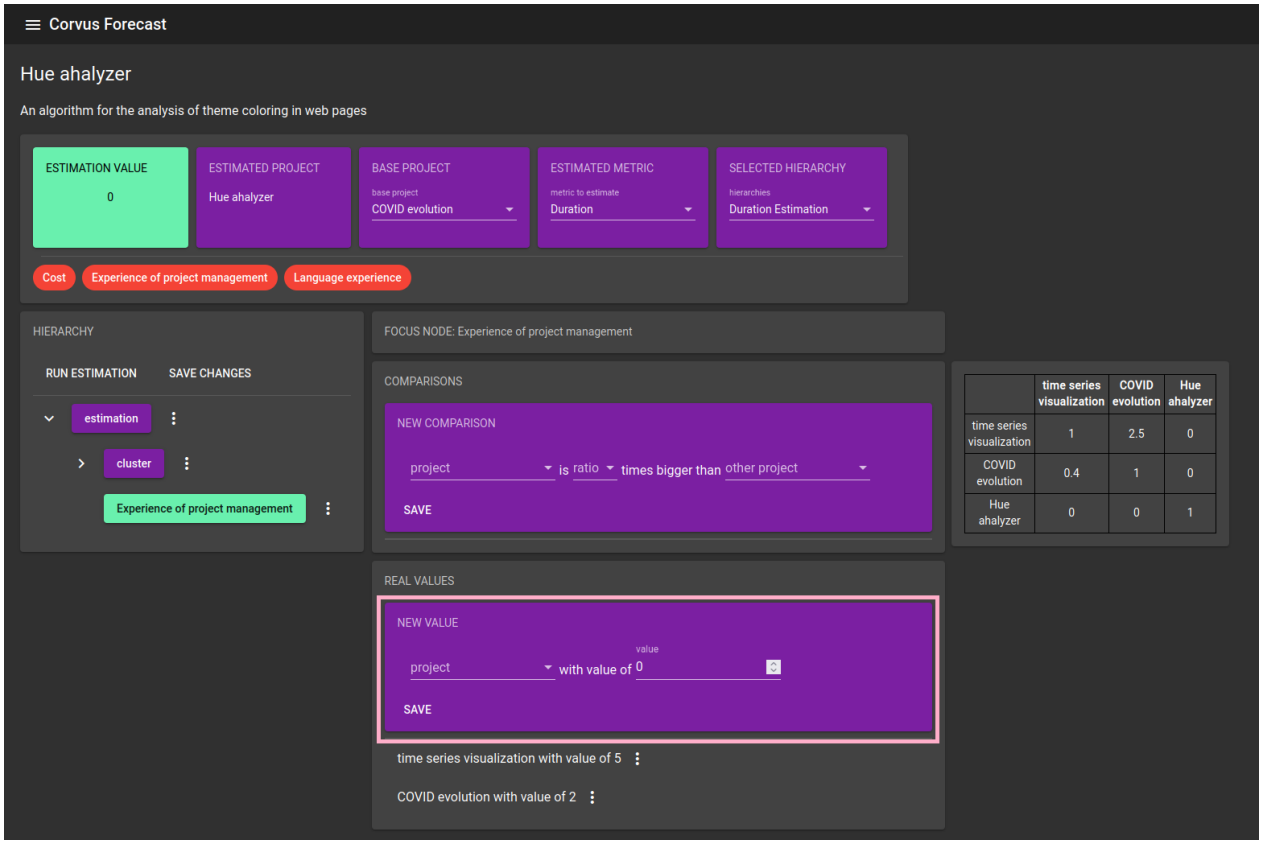


Partimos con la primera métrica, la experiencia en administración de proyectos.

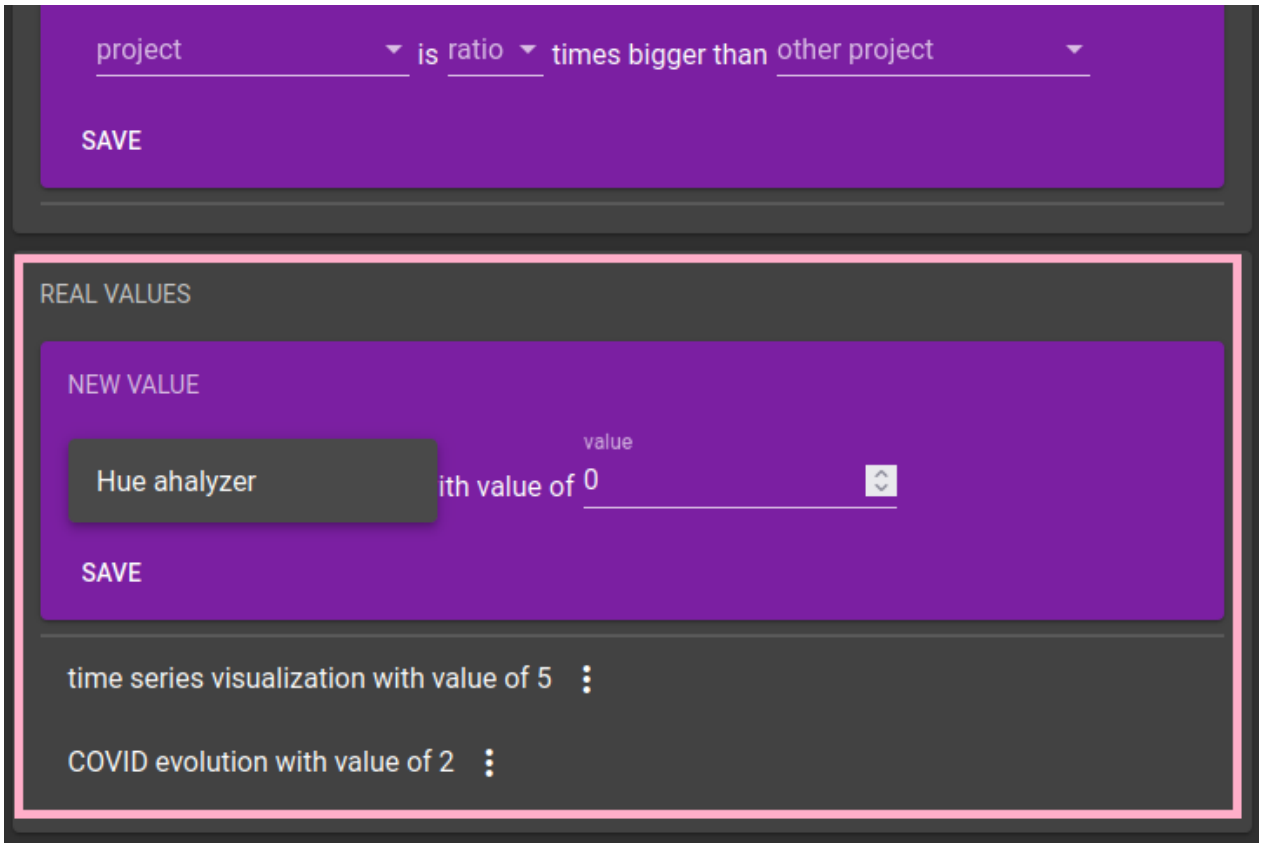


En la mitad inferior derecha vemos dos secciones, vemos dos partes, la primera es para ingresar comparaciones, la segunda es para ingresar valores reales. Por último a la derecha de todo se muestra una matriz representativa de la información conjunta de las comparaciones y valores reales.

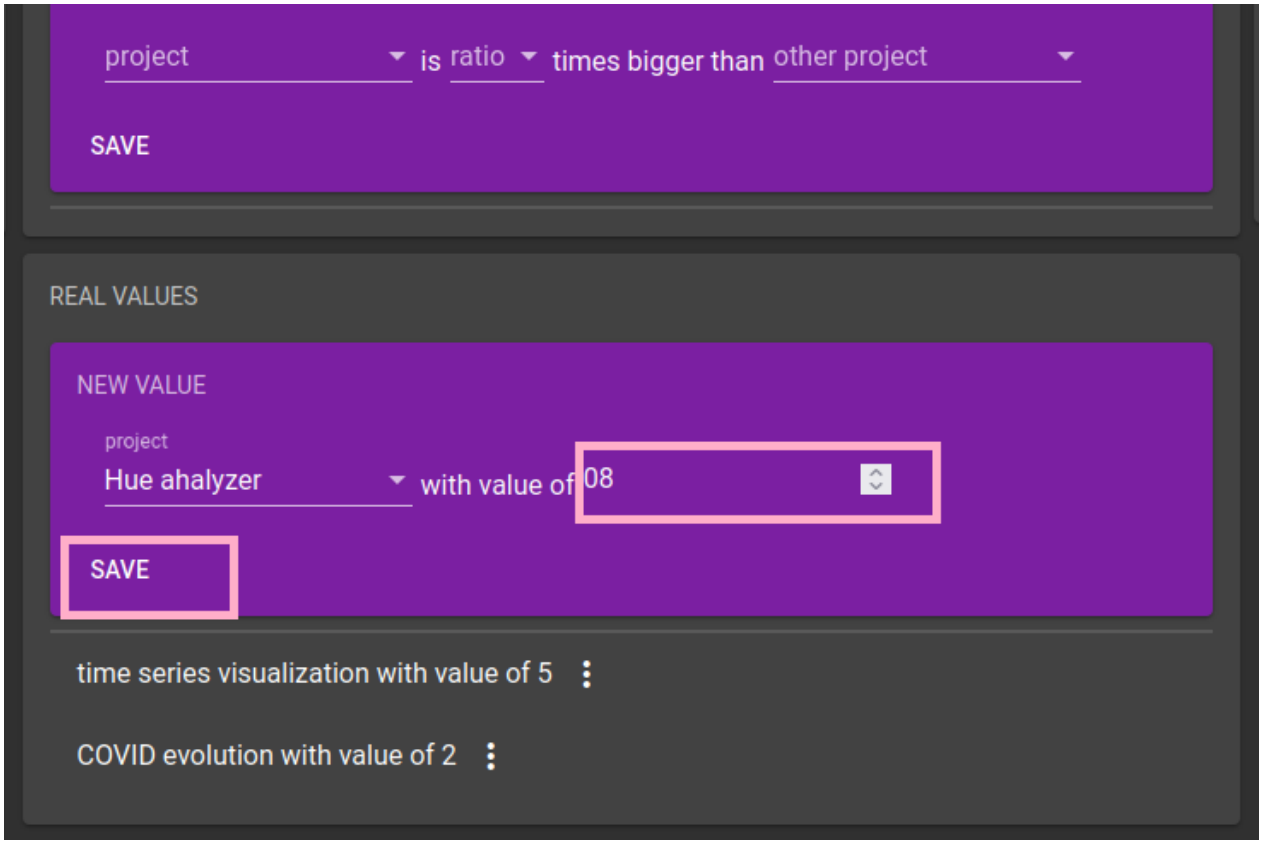
En esta ocasión sabemos el valor real de la experiencia en administración de proyectos, por lo que nos centramos en esta sección.



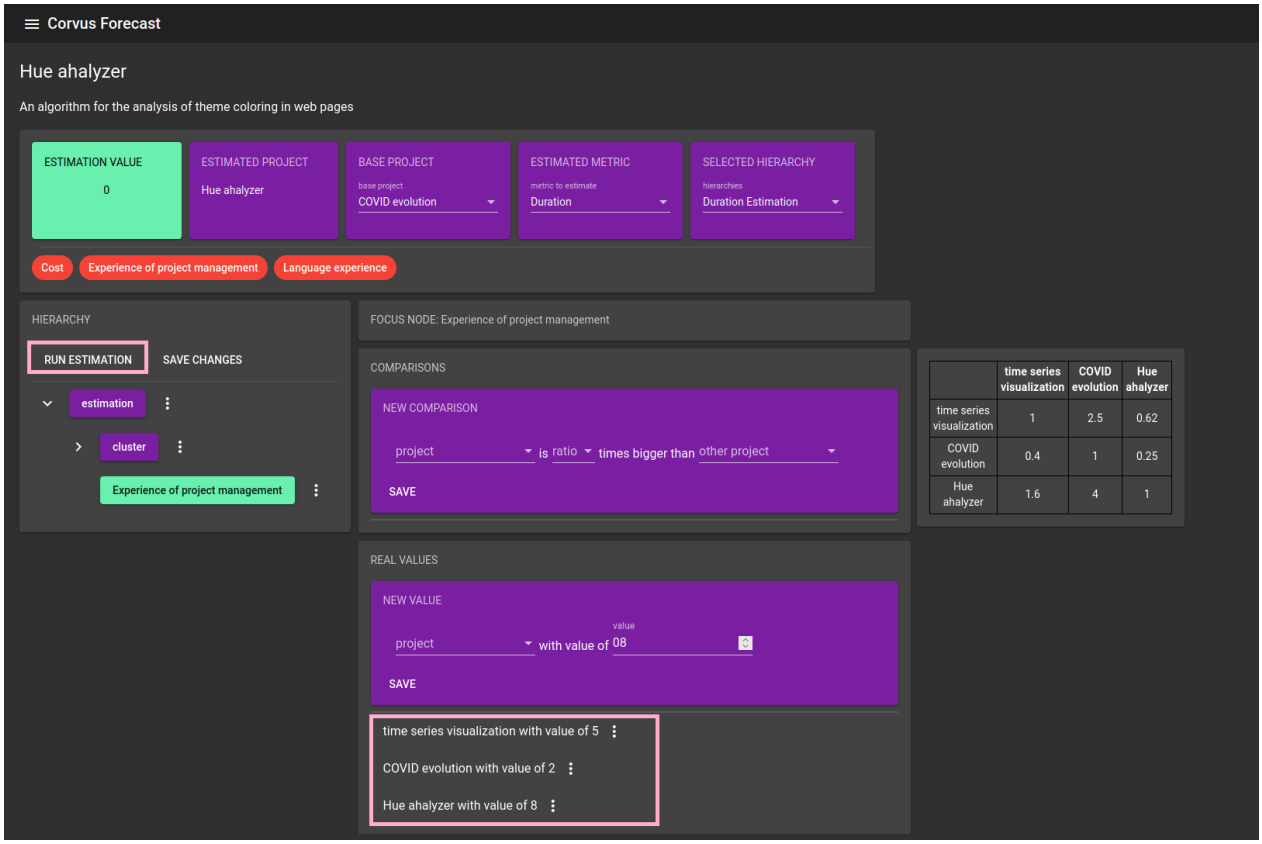
Elegimos el proyecto al cual ingresar el valor



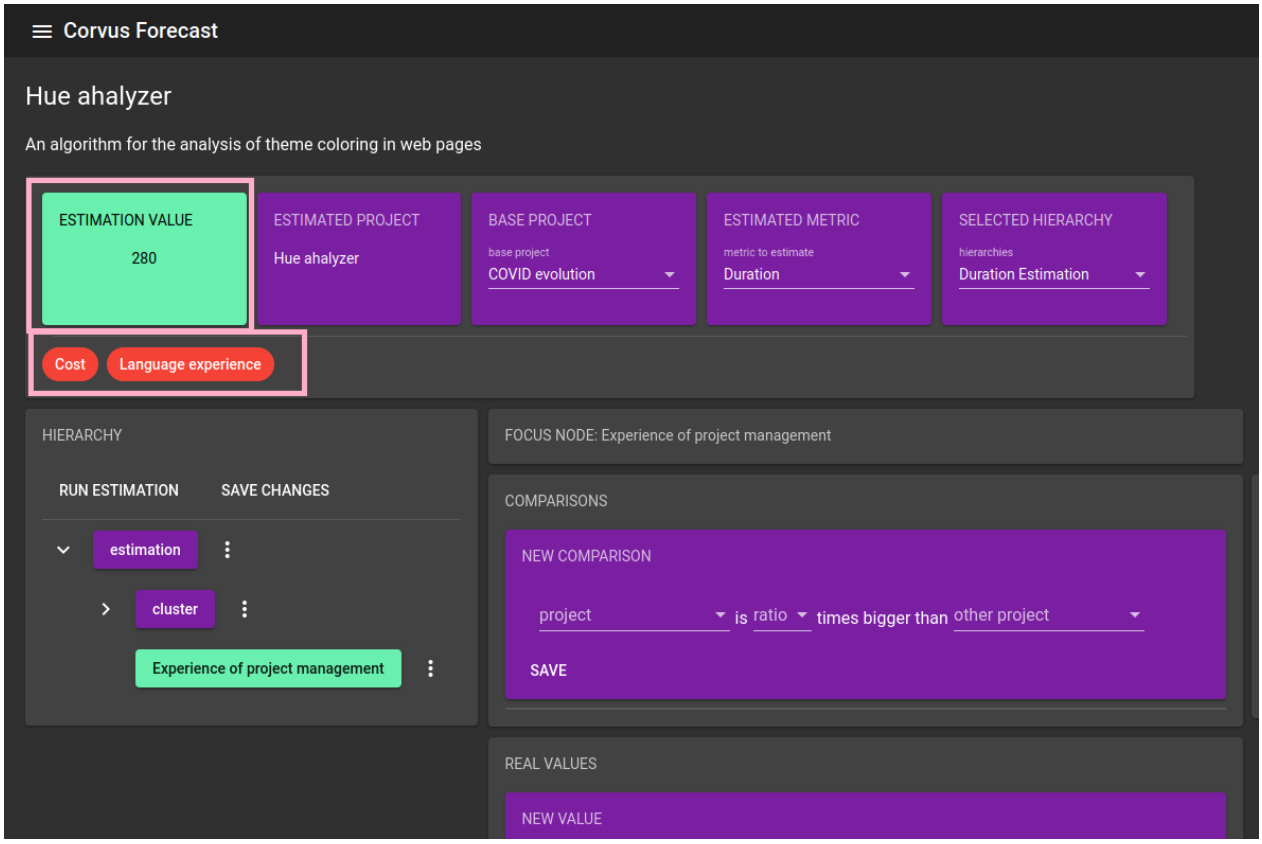
ingresamos el nuevo valor y guardamos.



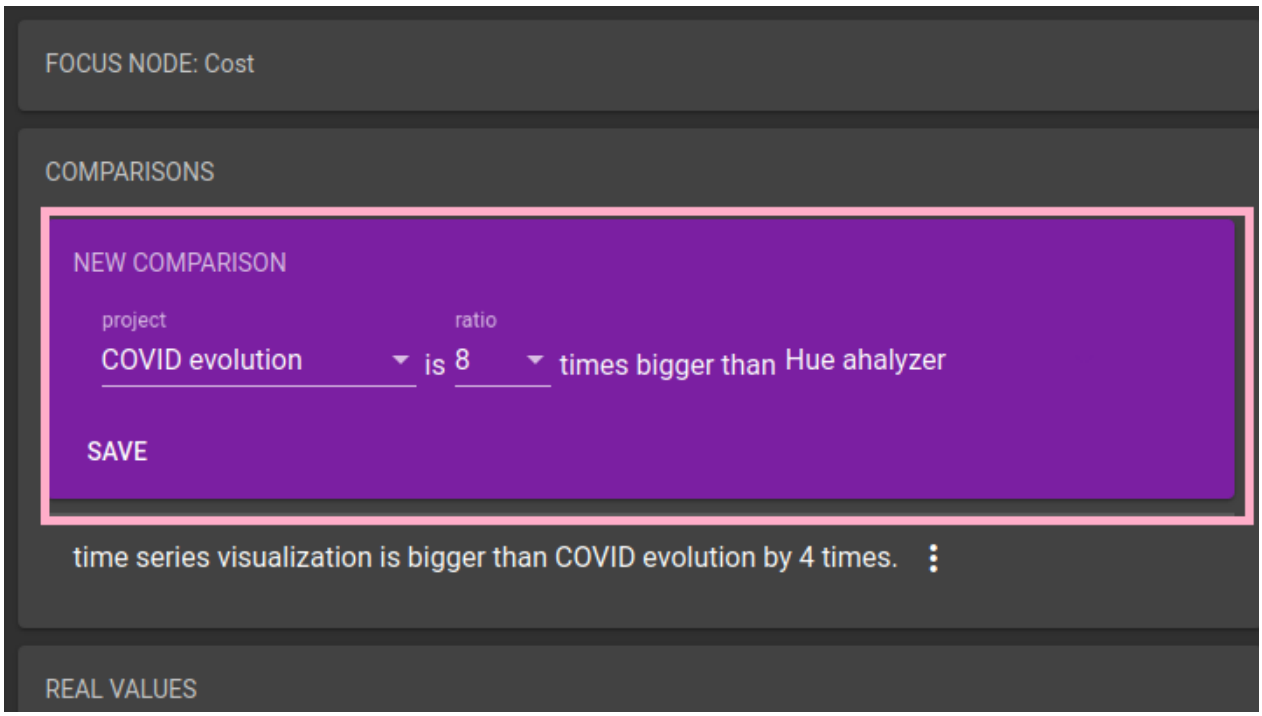
Ahora que tenemos valores reales para todos los proyectos de la métrica, volvemos a correr la estimación.



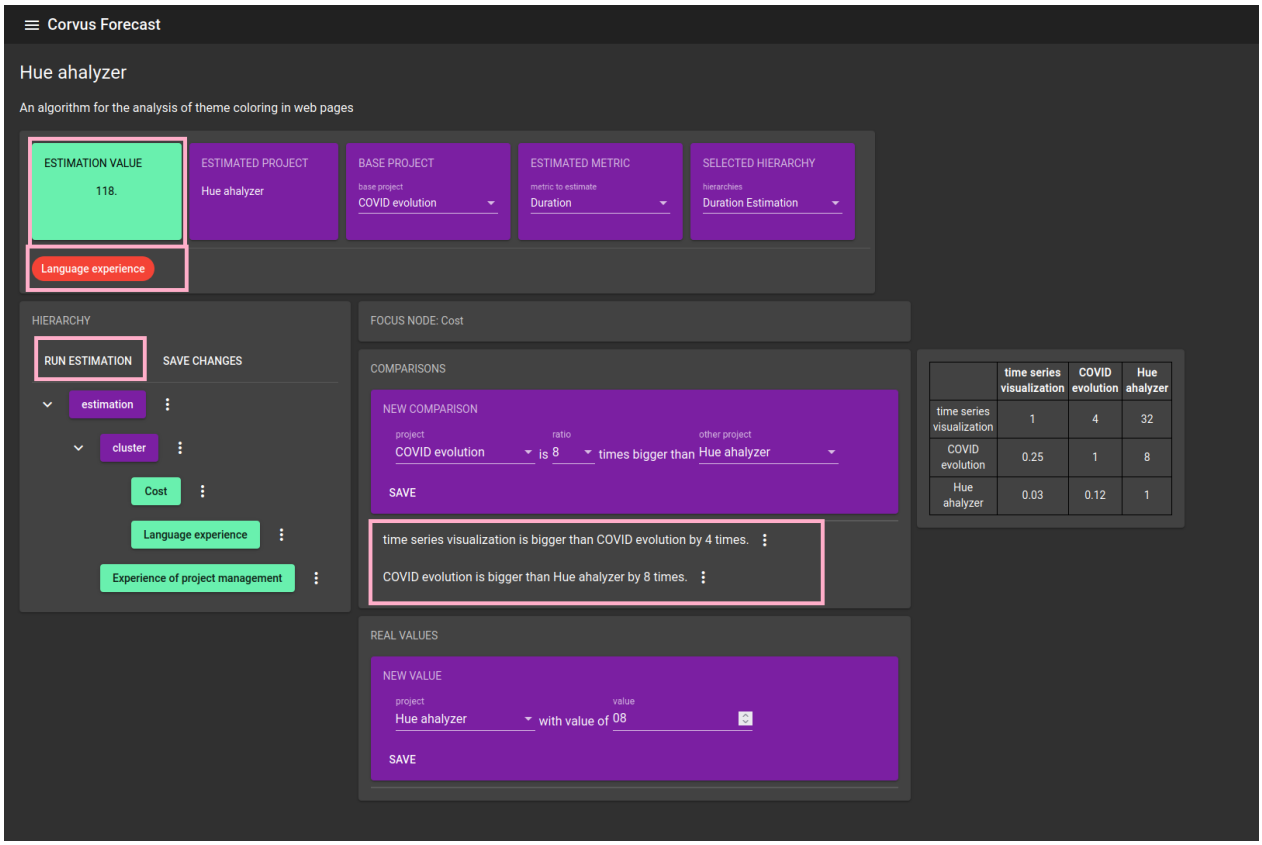
Vemos que todavía tenemos errores para otras dos métricas que requieren clausura, pero esta vez tenemos una estimación. Estimación parcial que no incluye las métricas sin clausura.



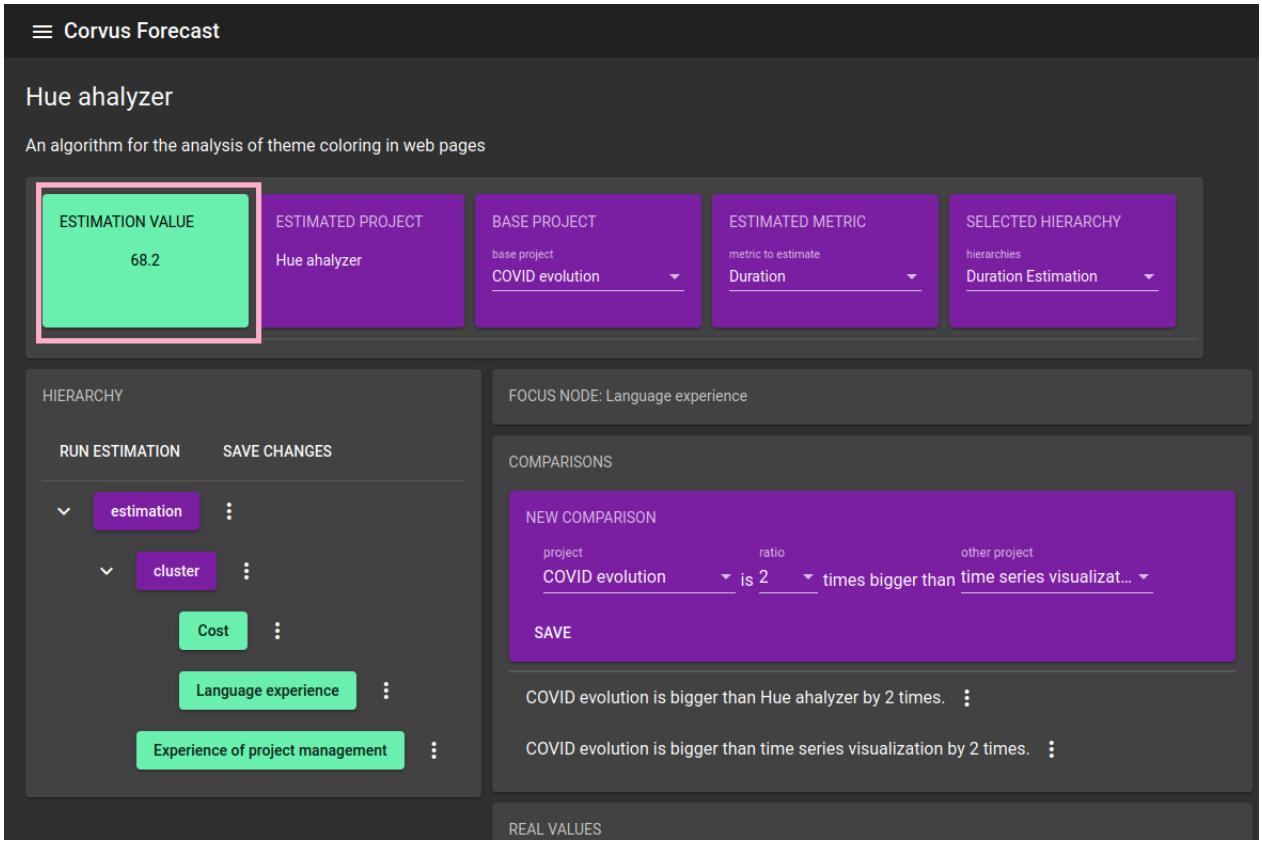
Lo siguiente que hacemos es completar la clausura en la métrica de Costo. Esta vez ingresamos una comparación entre el proyecto COVID evolution con el proyecto siendo estimado, el HUE analyzer. El proyecto COVID evolution es cerca de 8 veces el tamaño del proyecto HUE analyzer, así que ingresamos dicha información.



vemos la actualización de la lista de comparaciones, por lo que volvemos a correr la estimación. La estimación se actualiza, y solo nos queda por completar la clausura de la última métrica, la experiencia en el lenguaje de programación.



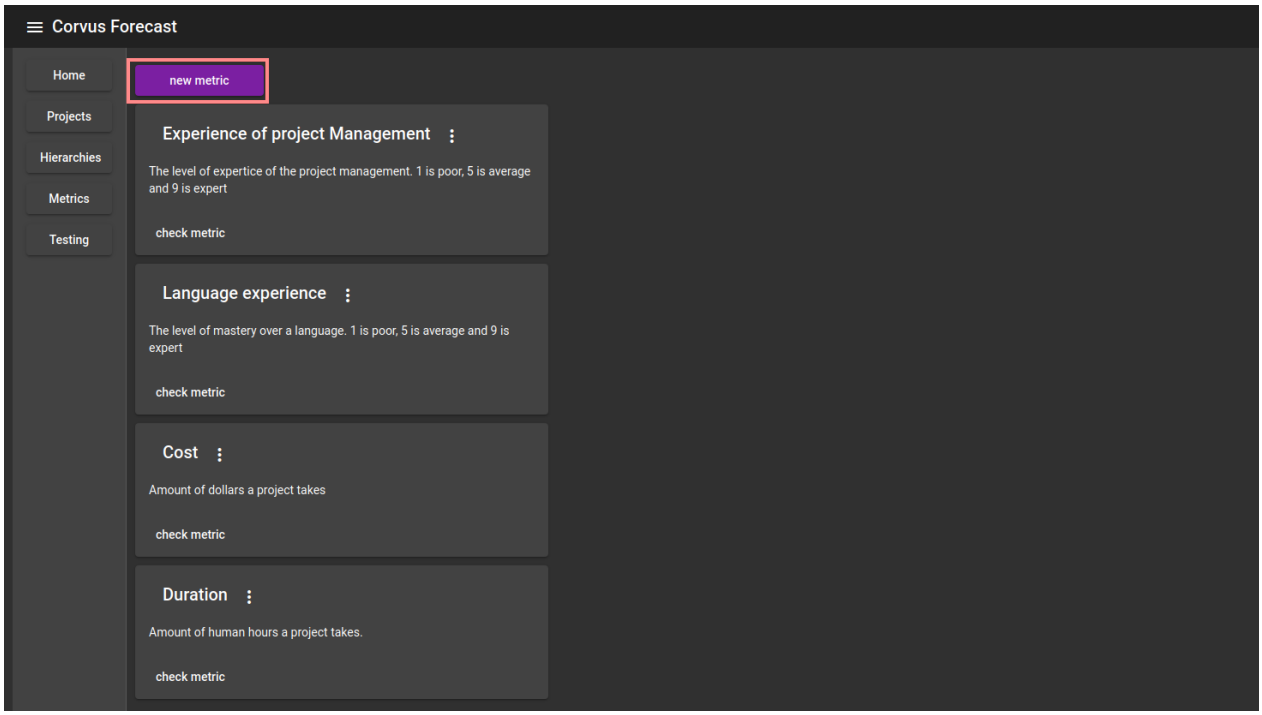
completamos la última métrica y volvemos a correr la estimación, consiguiendo la última versión de la estimación, esta vez incluyendo la información de todas las métricas.


Corvus Forecast
Hue ahalyzer
An algorithm for the analysis of theme coloring in web pages
ESTIMATION VALUE
68.2
ESTIMATED PROJECT
Hue ahalyzer
BASE PROJECT
base project
COVID evolution
ESTIMATED METRIC
metric to estimate
Duration
SELECTED HIERARCHY
hierarchies
Duration Estimation
HIERARCHY
RUN ESTIMATION
SAVE CHANGES
estimation
cluster
Cost
Language experience
Experience of project management
FOCUS NODE: Language experience
COMPARISONS
NEW COMPARISON
project
COVID evolution
is
ratio
2
times bigger than
other project
time series visualizat...
SAVE
COVID evolution is bigger than Hue ahalyzer by 2 times.
COVID evolution is bigger than time series visualization by 2 times.
REAL VALUES

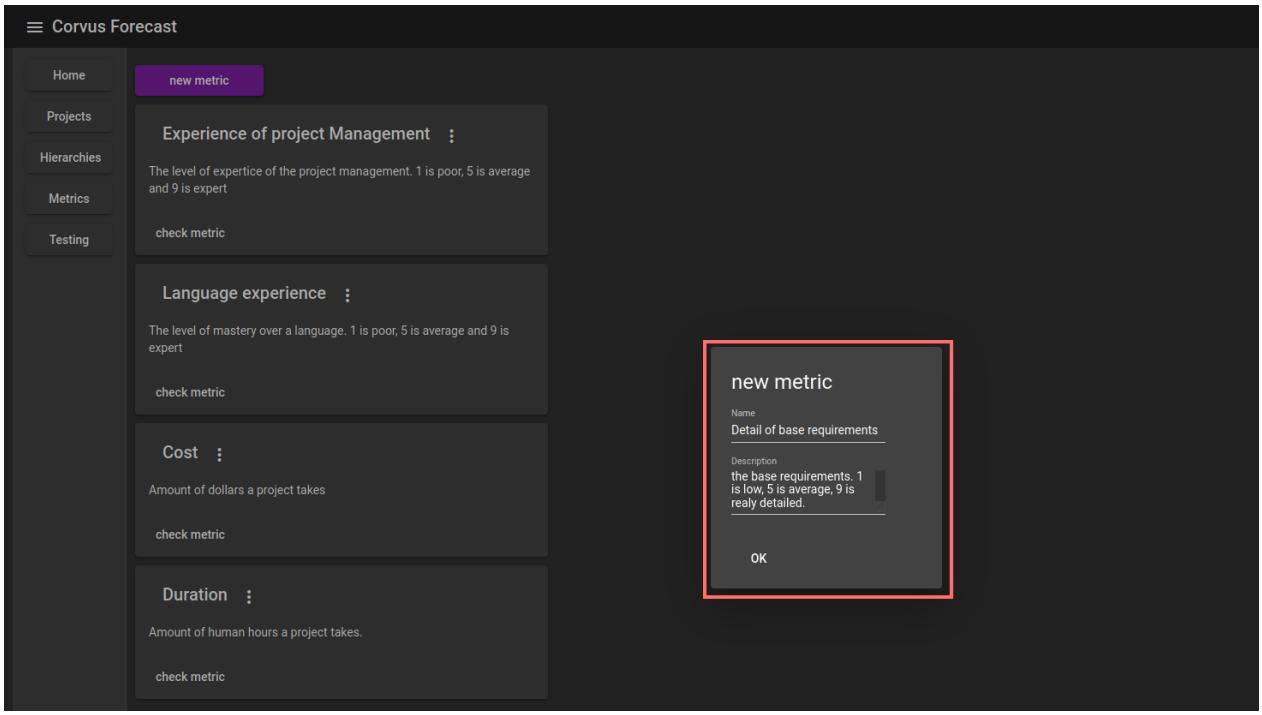
Caso de uso 2: Creación de una métrica

Lo que haremos en este ejemplo será crear una métrica, y agregar los valores reales de los proyectos presentes en la herramienta. Este ejemplo es la continuación del ejemplo anterior, así que contamos con tres proyectos y cuatro métricas.

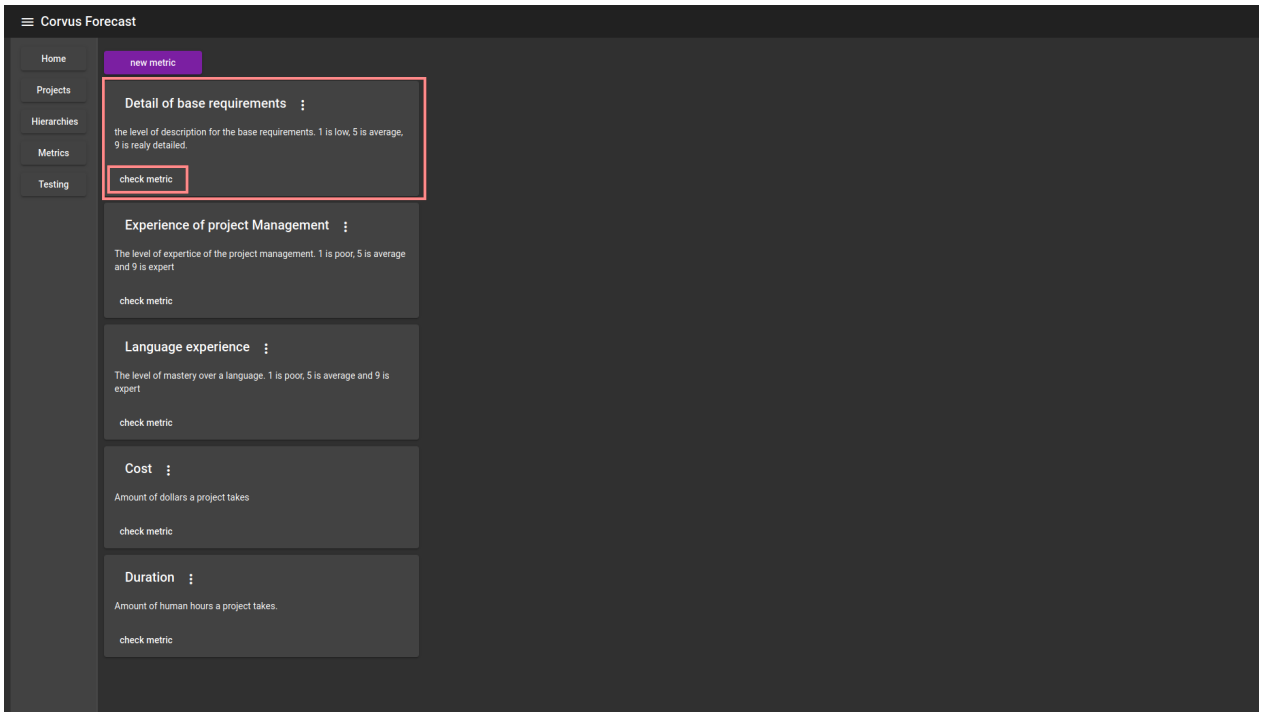
Partimos en la página de métricas, donde abrimos el diálogo para crear una nueva métrica.



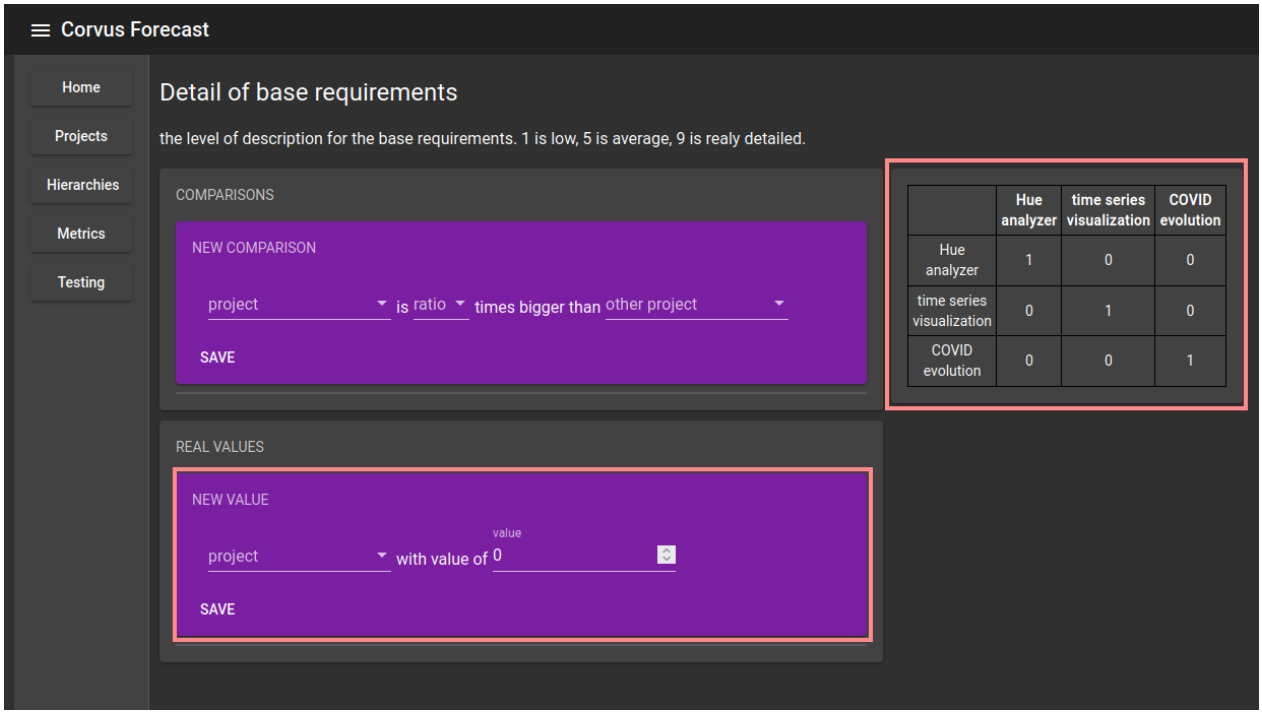
ingresamos su nombre y su descripción.



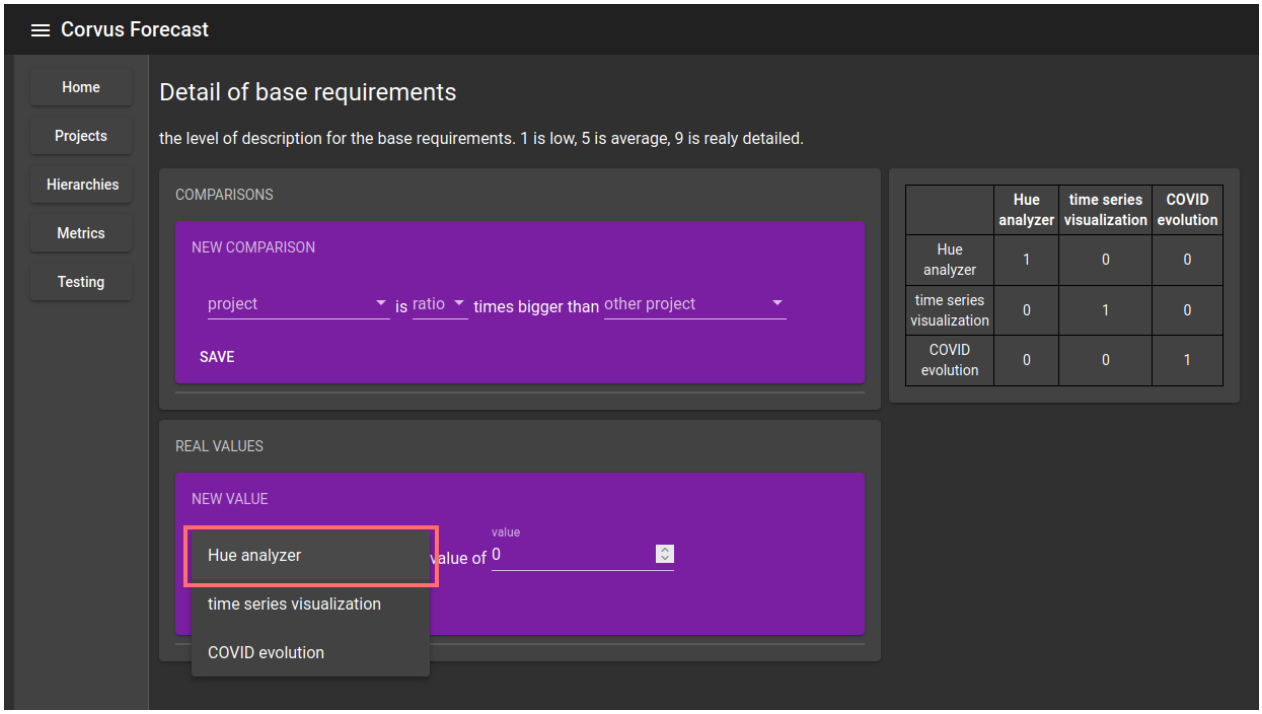
y la herramienta se actualiza, mostrando ahora la nueva métrica creada, "Detail of base requirements", o el detalle de los requerimientos base, una métrica que refleja el nivel de detalle de los requerimientos de un proyecto, 1 es un detalle bajo, 5 es promedio y 9 es un detalle alto. Proseguimos a ver la página de la métrica.



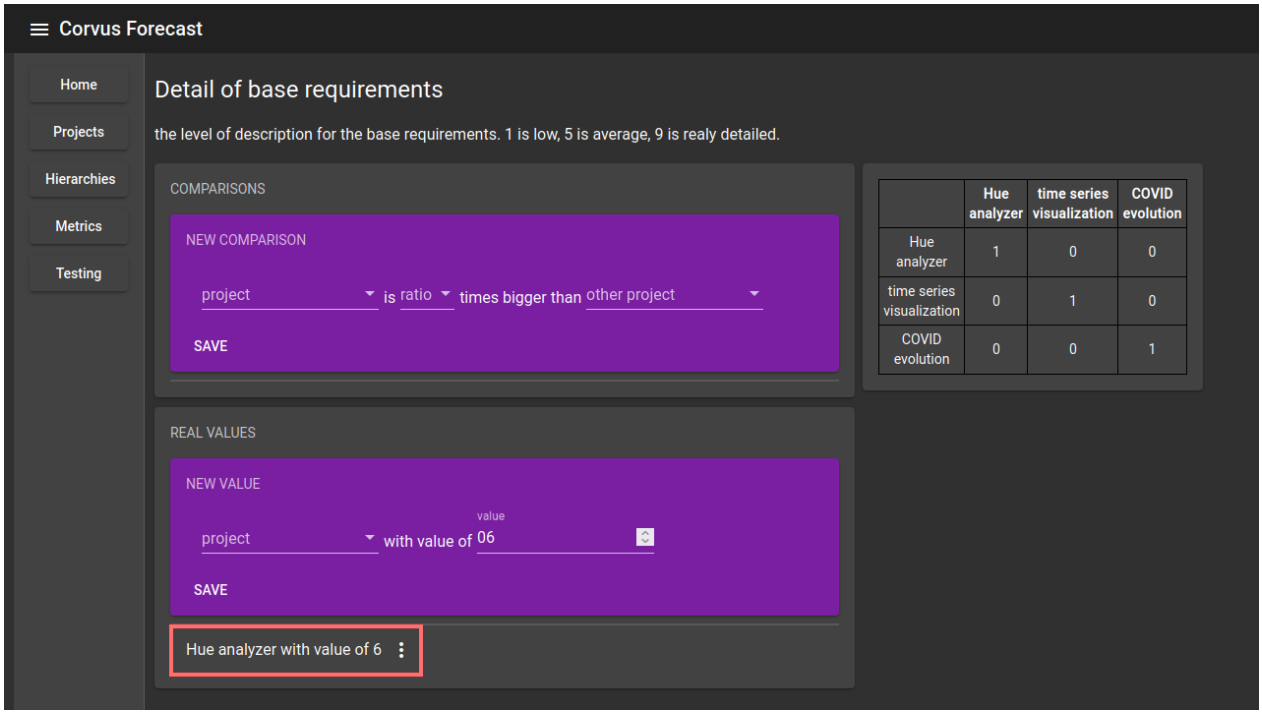
La página de la métrica posee la misma interfaz de usuario que la presentada en los nodos jerárquicos. A la derecha podemos ver la matriz de comparaciones de la métrica, que se irá actualizando a medida que agreguemos información. En la parte inferior tenemos la sección para el ingreso de valores reales.



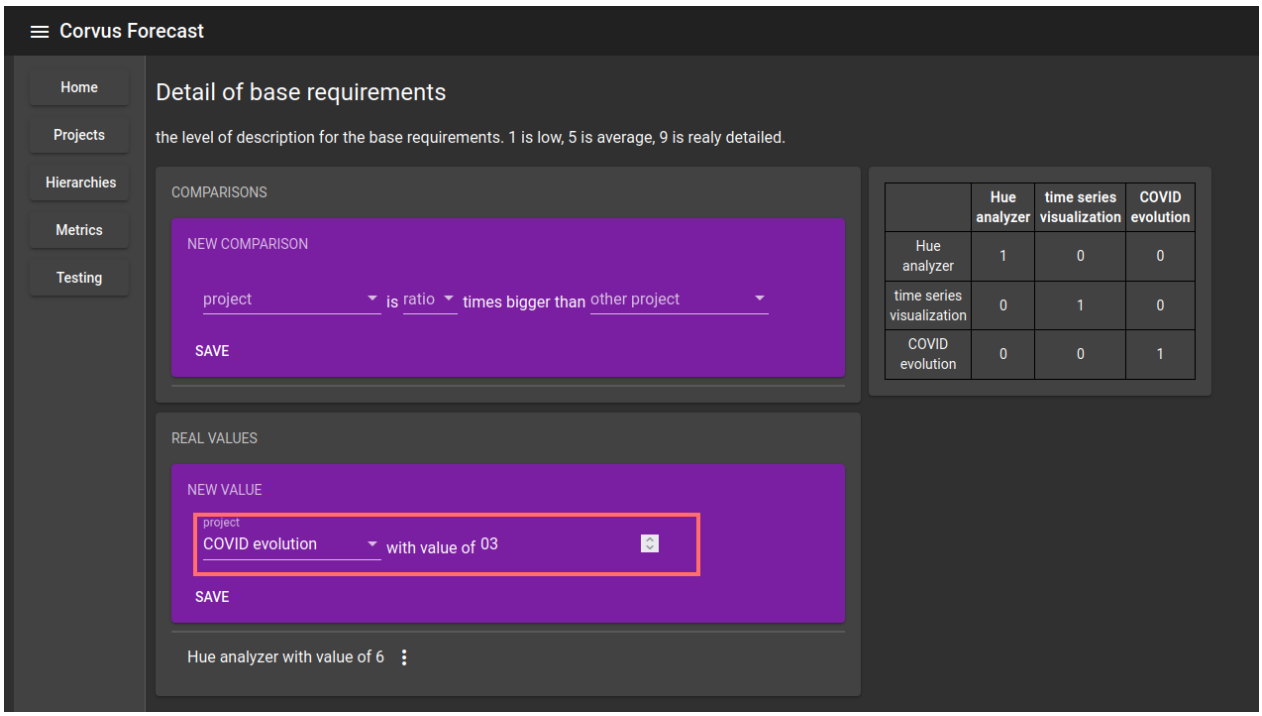
Seleccionamos un proyecto de la lista, y le asignamos un valor



La lista de proyectos con valores actuales se actualiza, y vemos el nuevo valor ingresado.

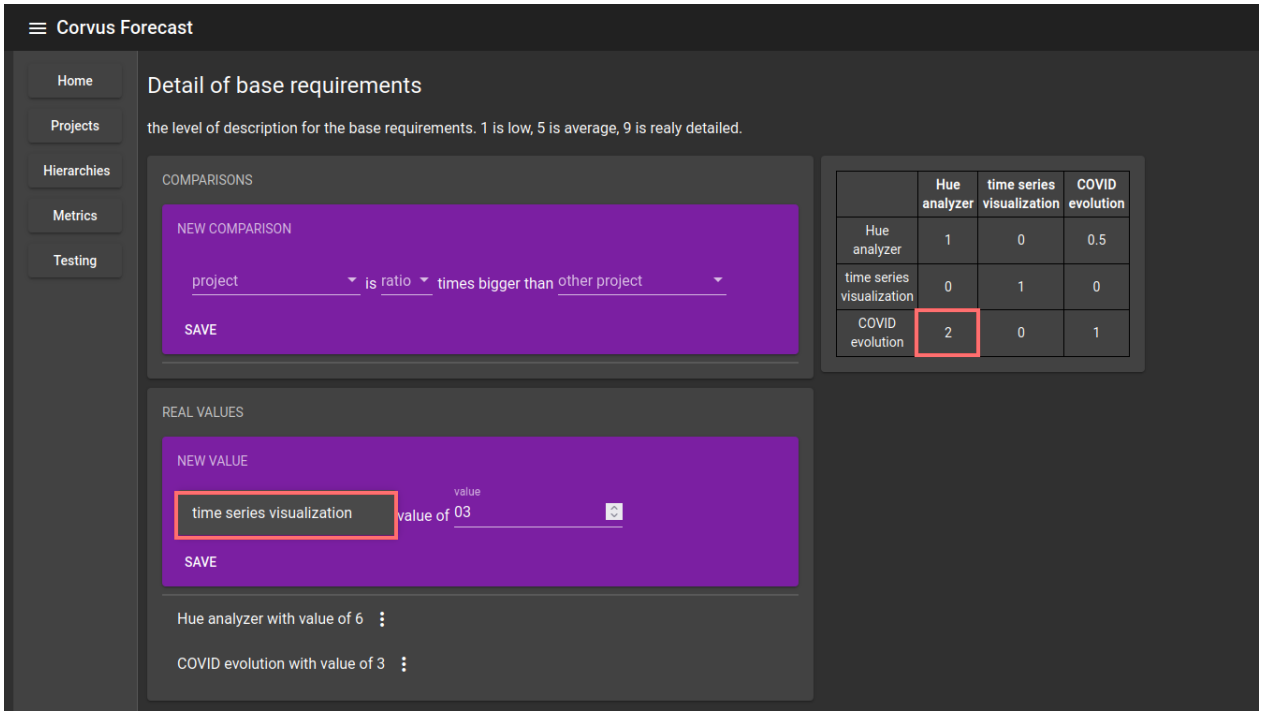


proseguimos con el siguiente proyecto

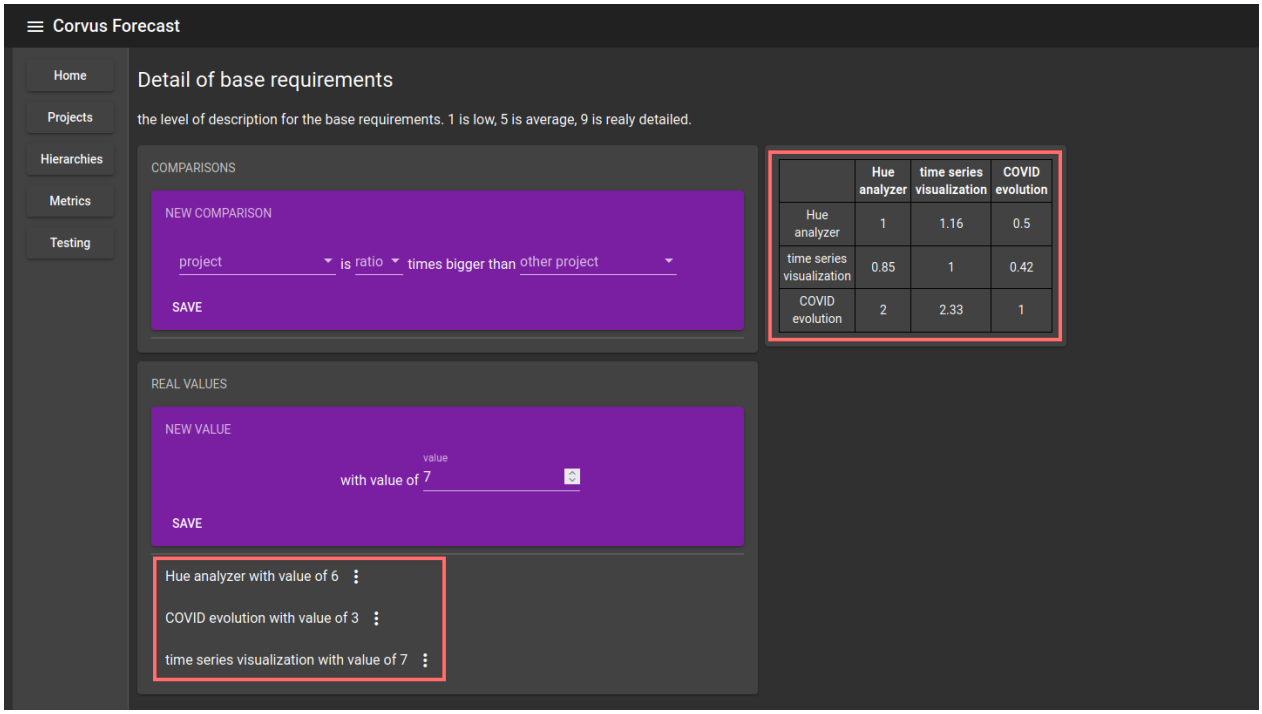


y con el último. Esta vez, la lista solo tiene un proyecto como opción, ya que es el único proyecto que queda sin valor asociado.

A la derecha, en la matriz, vemos que ya comenzamos a tener comparaciones a partir de estos valores.



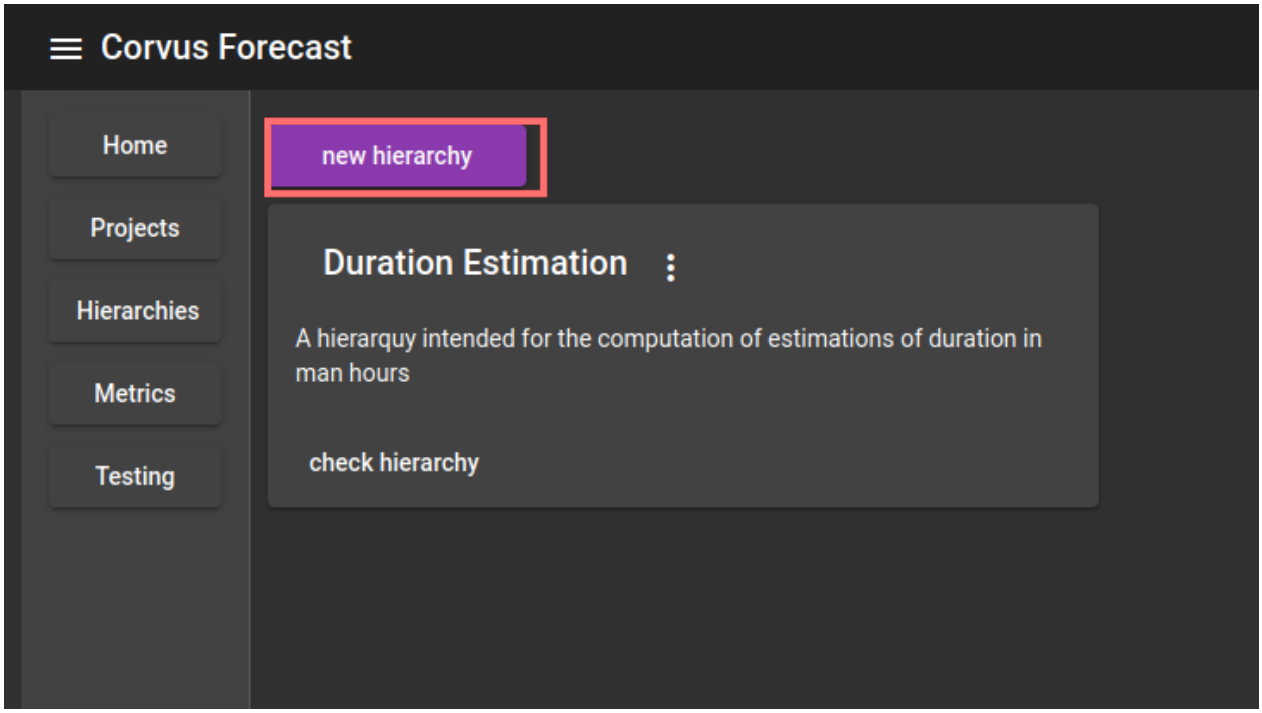
Una vez tenemos todos los proyectos con valores reales, vemos que la matriz de comparaciones está completa.



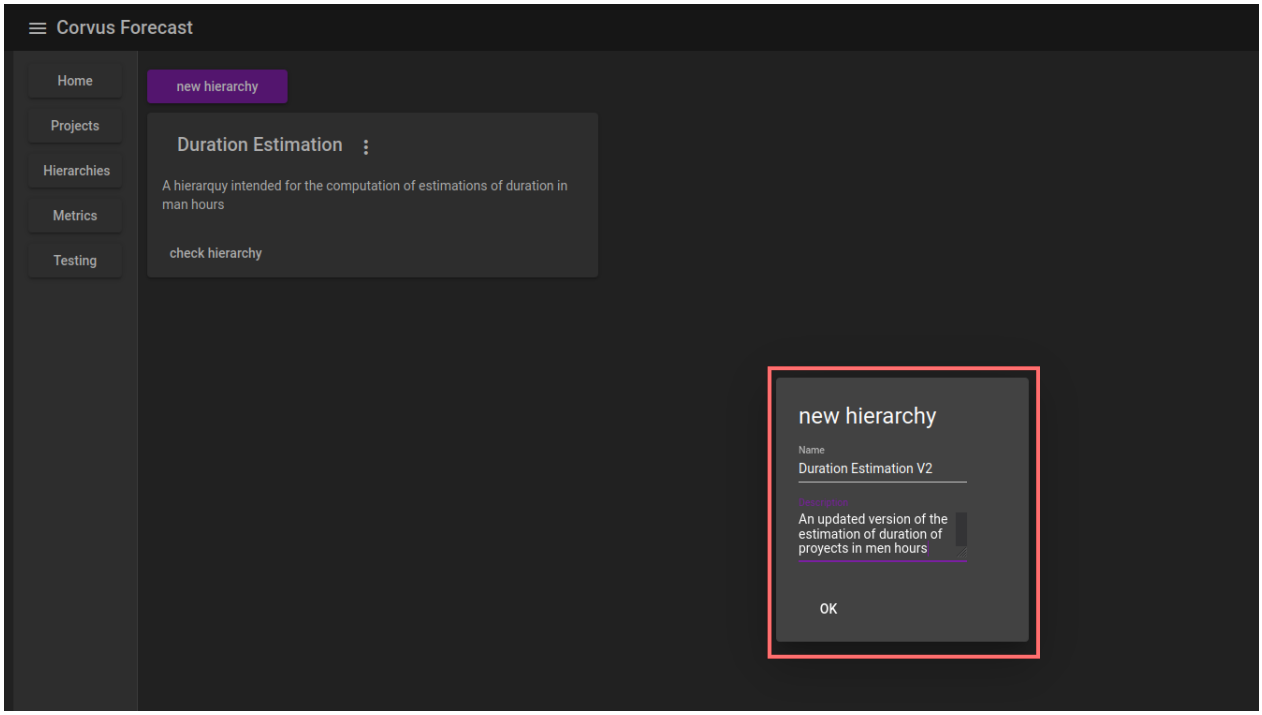
Caso de uso 3: creación de una jerarquía

En este ejemplo, veremos la creación de una nueva jerarquía, para seguir haciendo estimaciones de duración, como la jerarquía ya presente en la herramienta, pero esta vez incluiremos la nueva métrica creada en el ejemplo anterior, haciendo uso efectivo de todas las métricas en la herramienta.

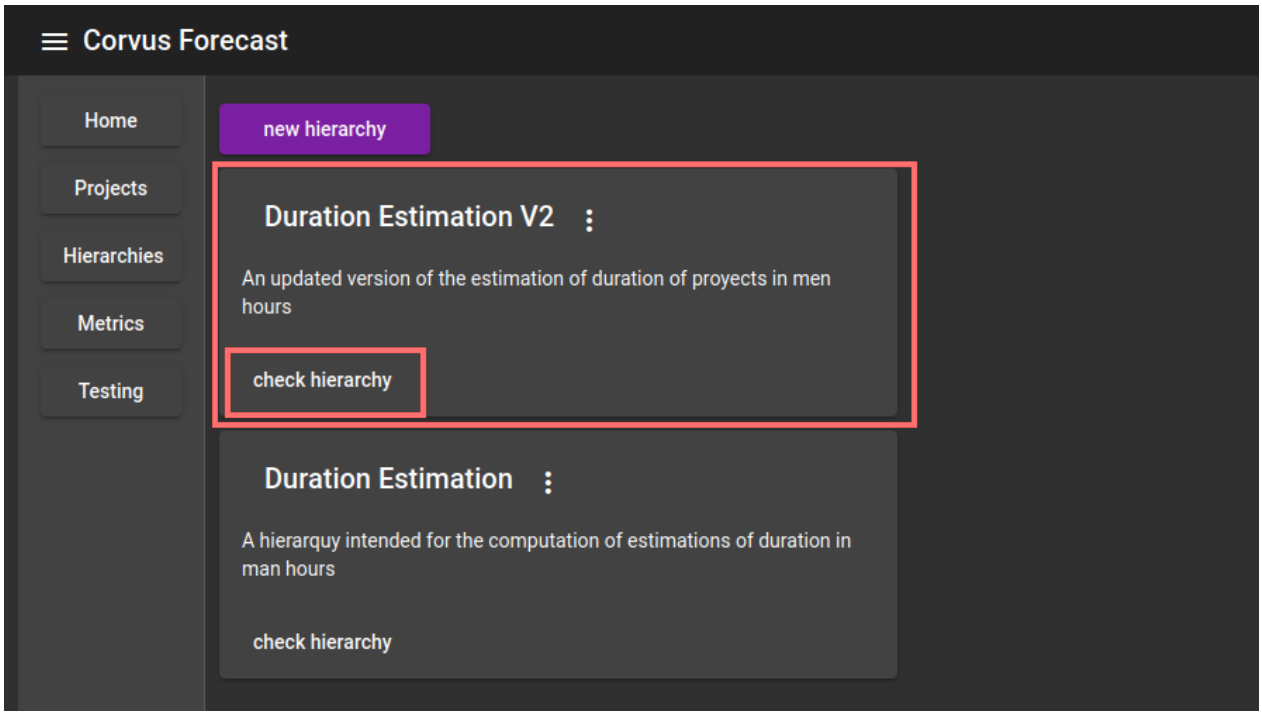
Partimos en la página de las jerarquías, y procedemos a abrir el diálogo de creación de una nueva jerarquía.



Le asignamos un nombre y una descripción a la nueva jerarquía

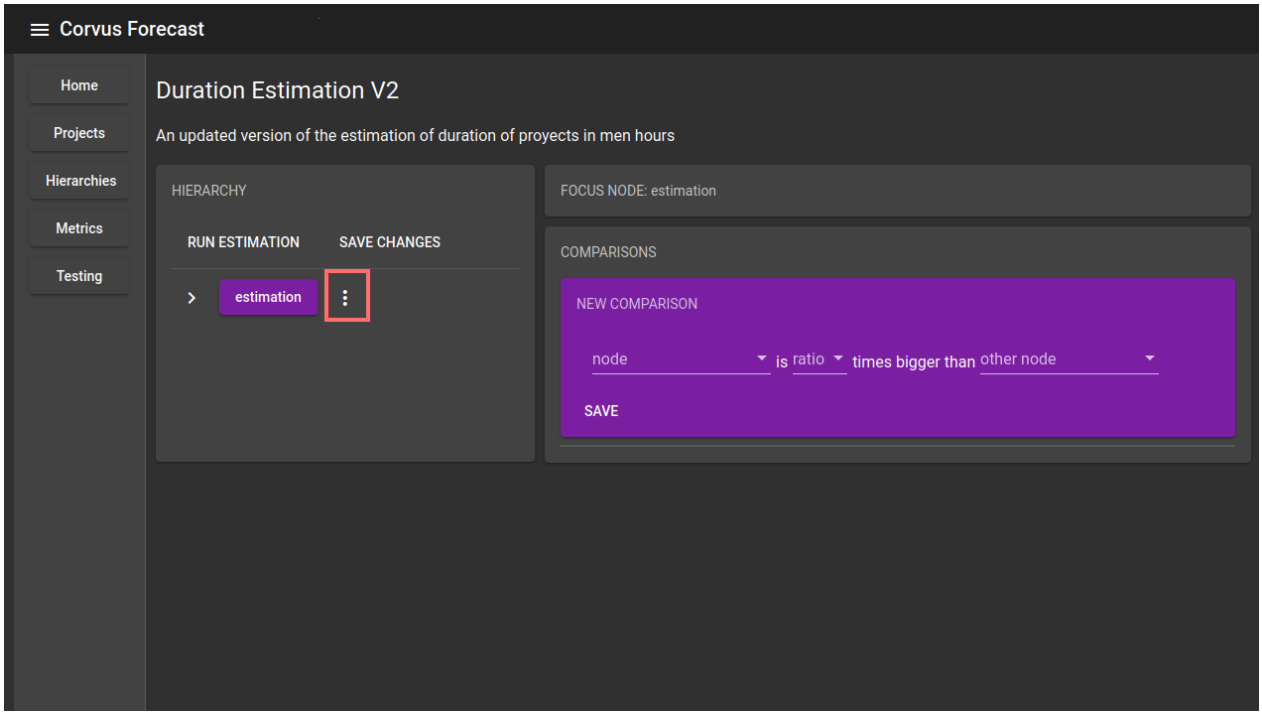


Esta nueva jerarquía, "Duration Estimation V2", o versión 2 de la estimación de duración, es la versión actualizada de la jerarquía anterior. Procedemos a la página de la jerarquía, para modificarla.

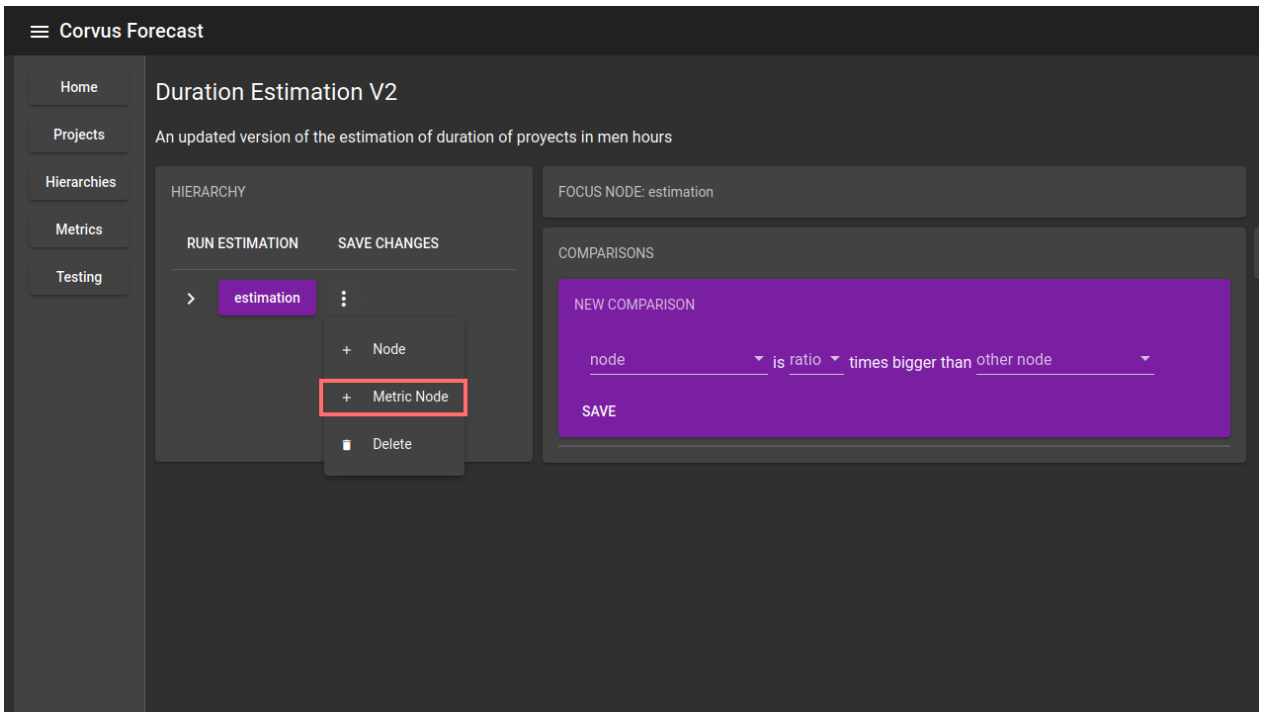


En la página de la jerarquía, a la izquierda vemos la estructura jerárquica de la jerarquía, mientras que a la derecha vemos las comparaciones del nodo en foco.

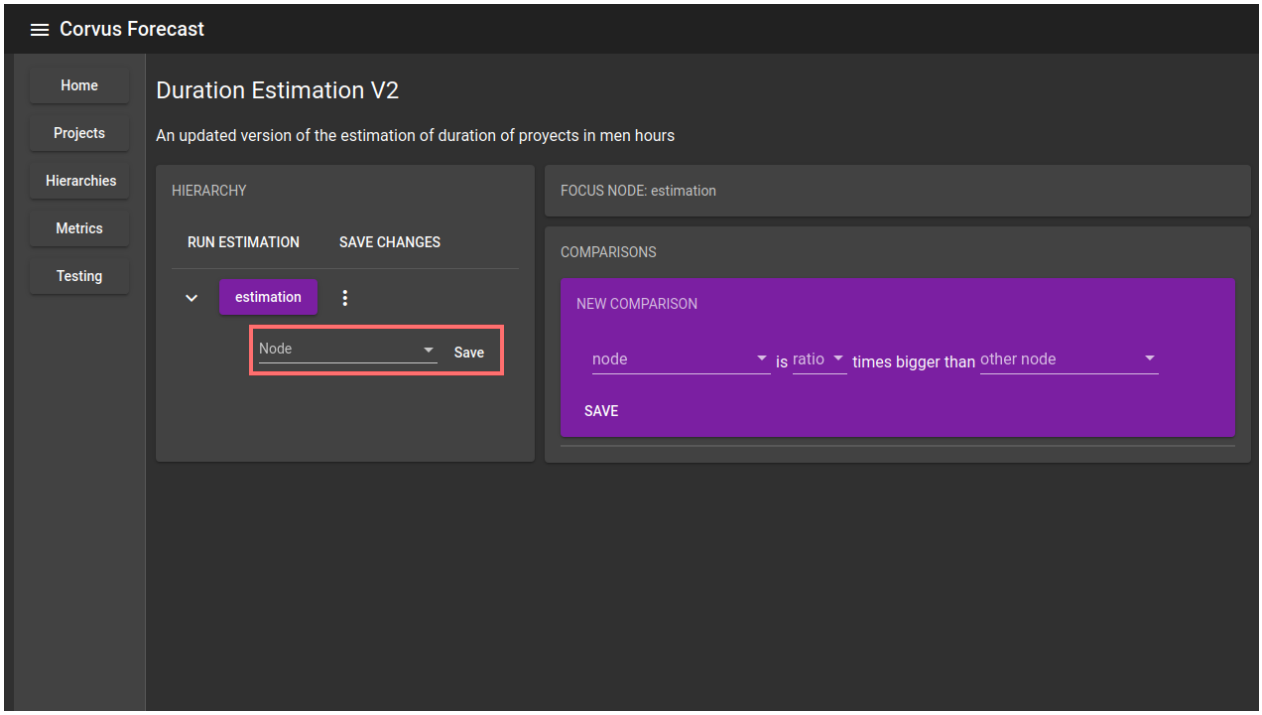
Como la jerarquía está recién creada posee solo un nodo. El siguiente paso es extenderla. Abrimos el menú asociado al nodo.



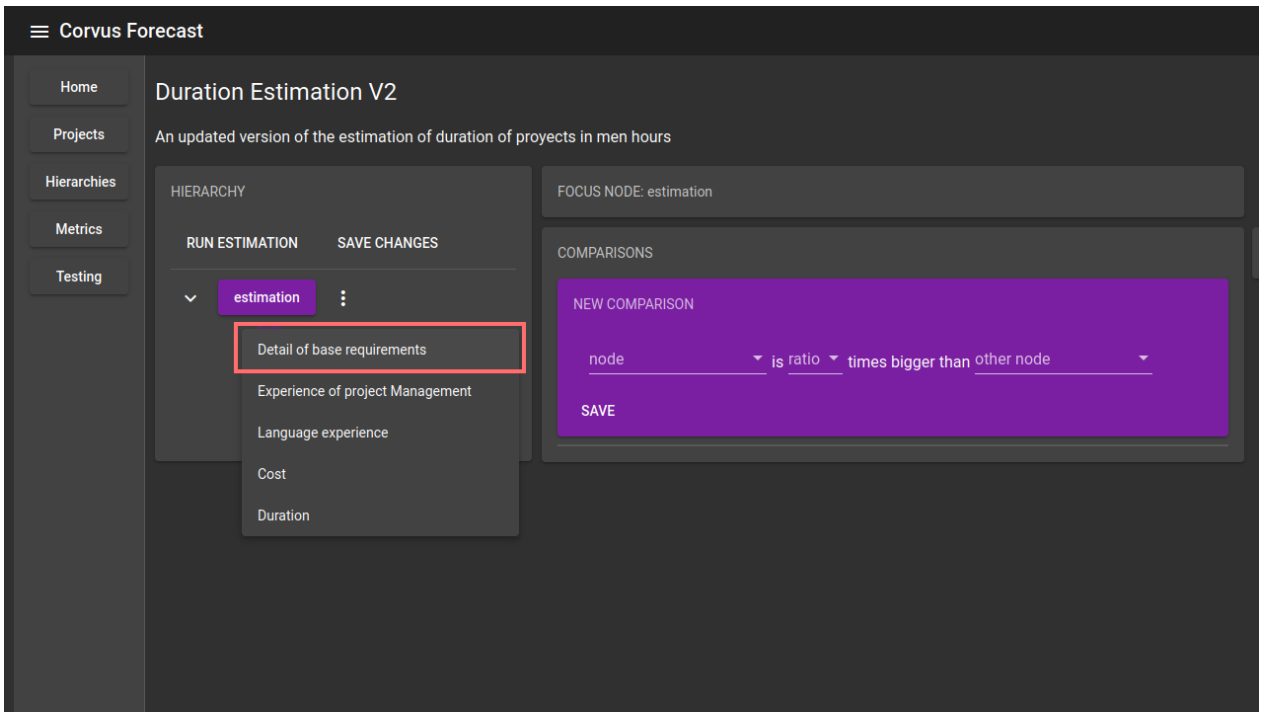
Y escogemos agregar un nuevo nodo métrico.



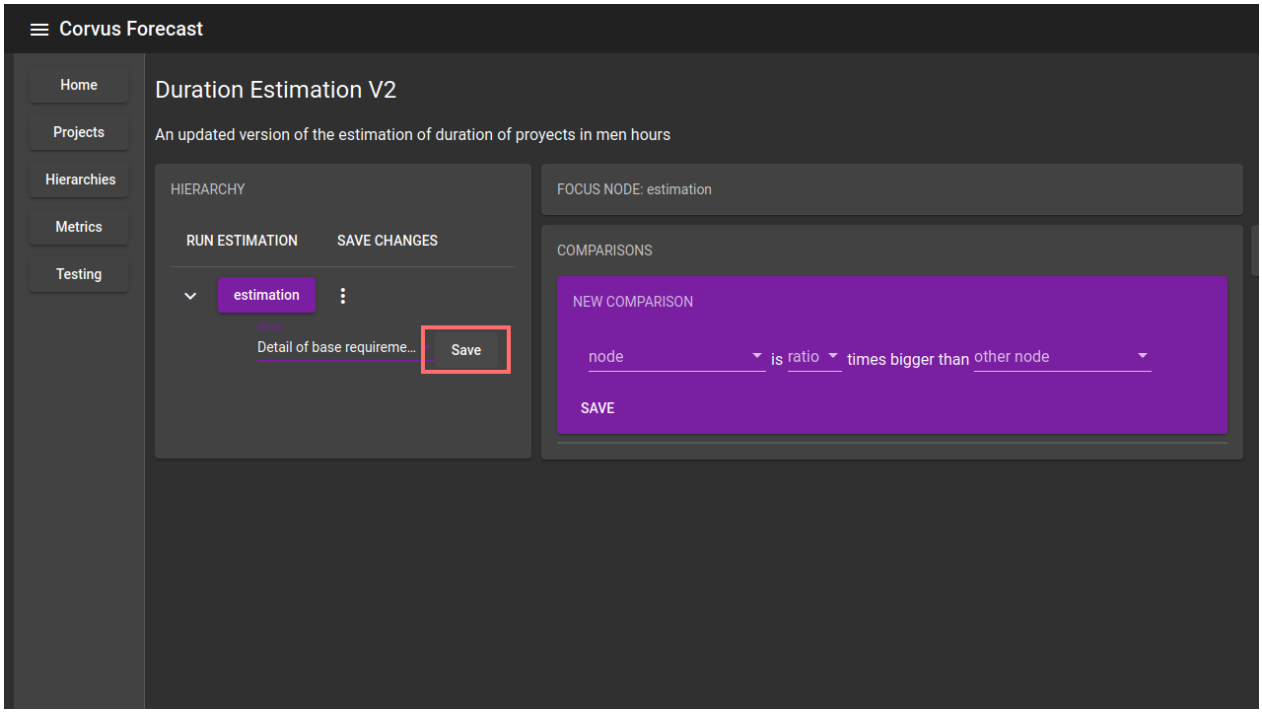
La jerarquía se actualiza y nos pide elegir cuál métrica queremos utilizar para el nuevo nodo.



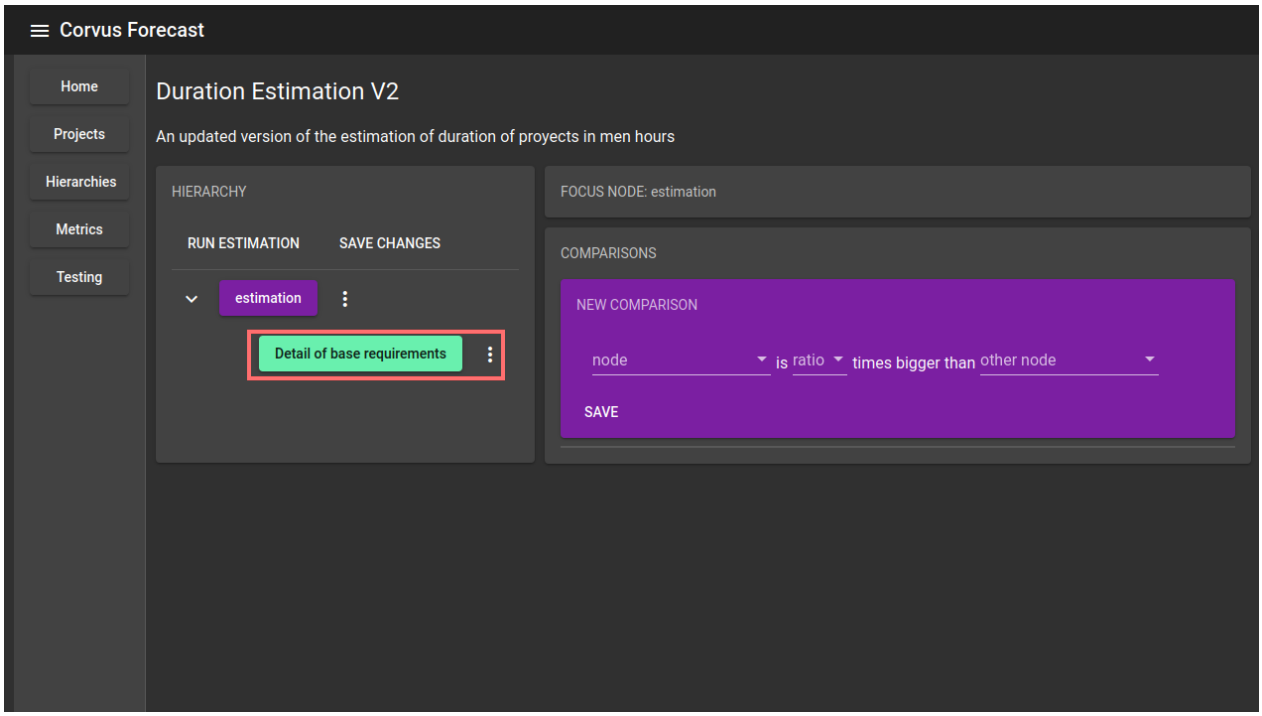
elegimos la nueva métrica que creamos en el ejemplo anterior.



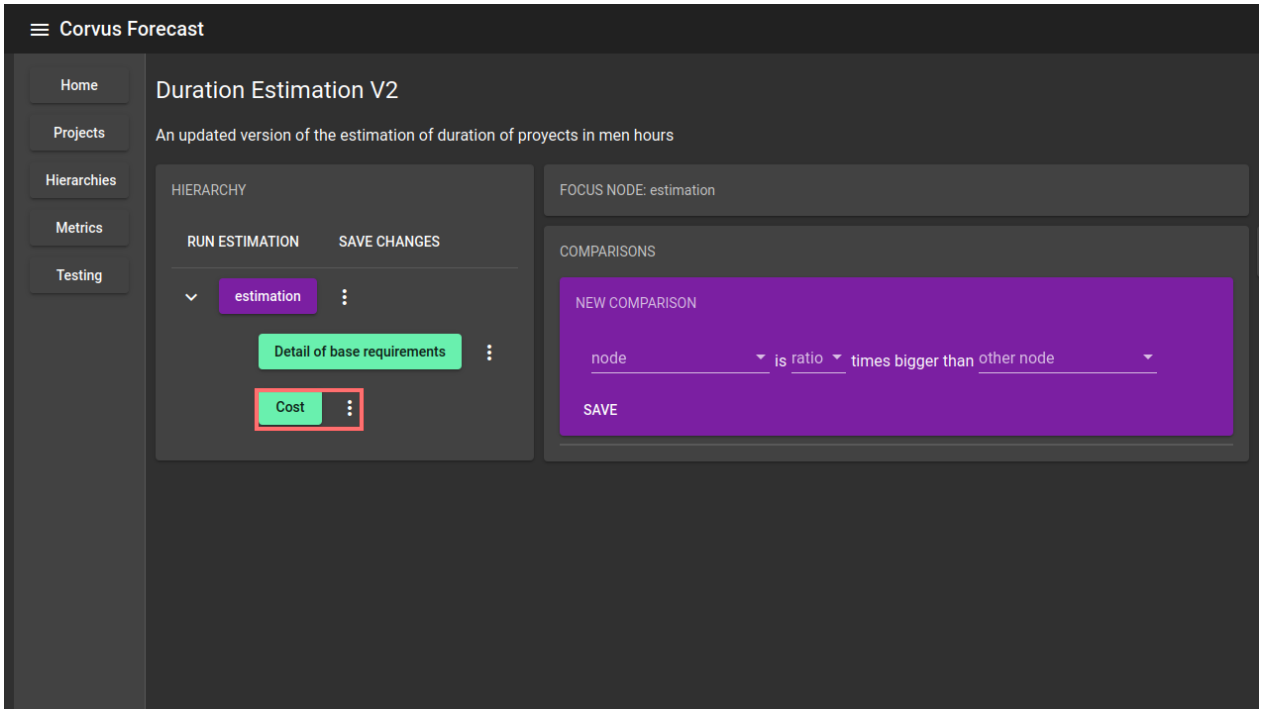
y guardamos el cambio.



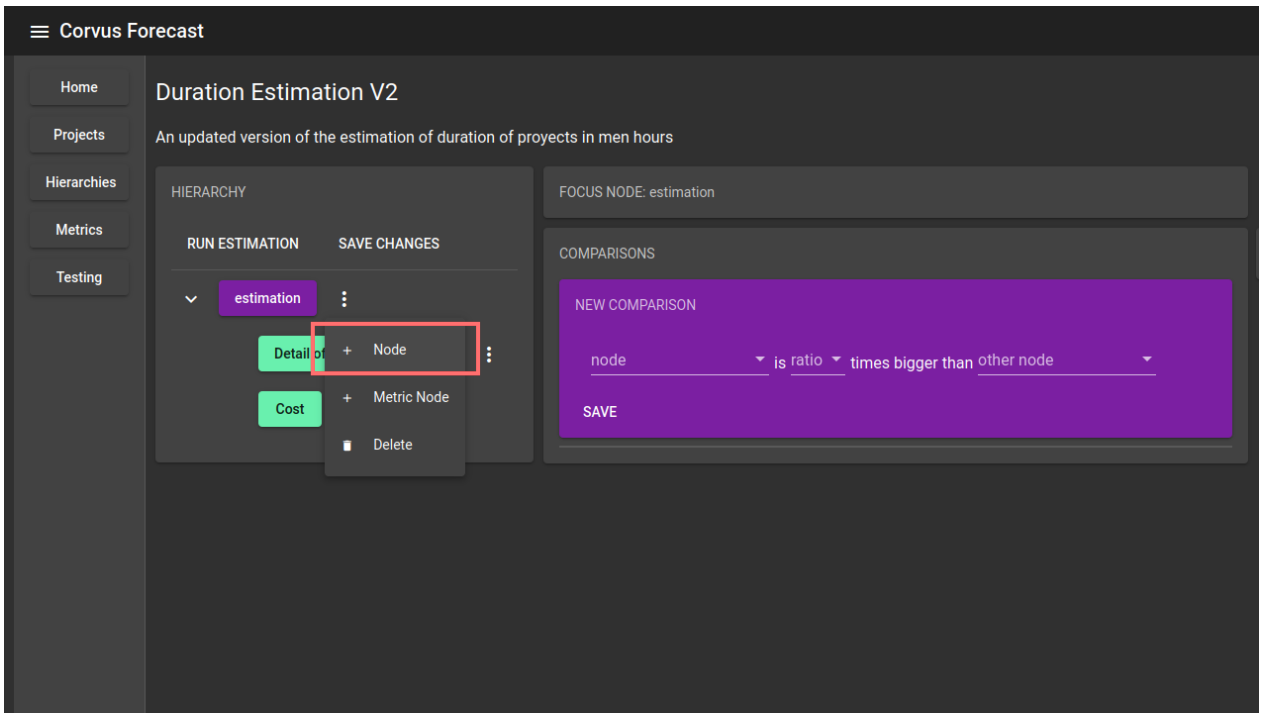
La nueva métrica ahora es parte de la jerarquía. Como se ve en la imagen, los nodos jerárquicos son morados, mientras que los nodos métricos son verdes.



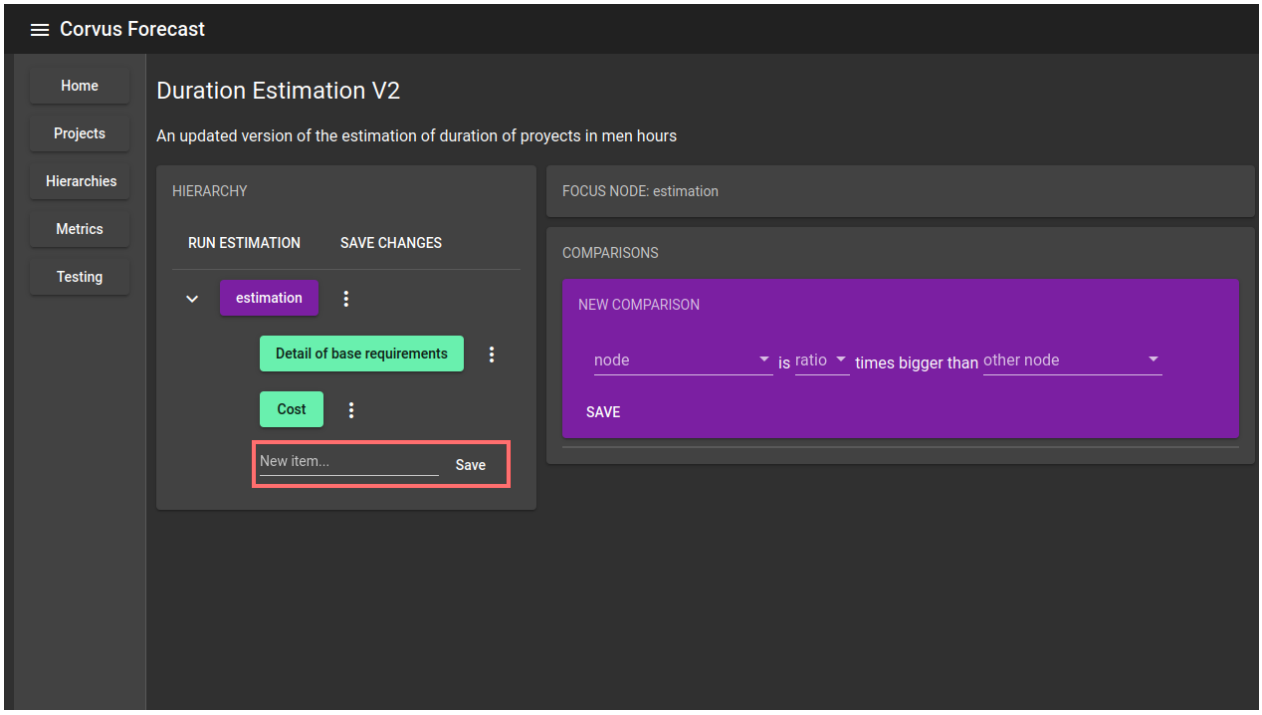
Repetimos los pasos para agregar un nodo métrico, y esta vez incluimos la métrica de coste.



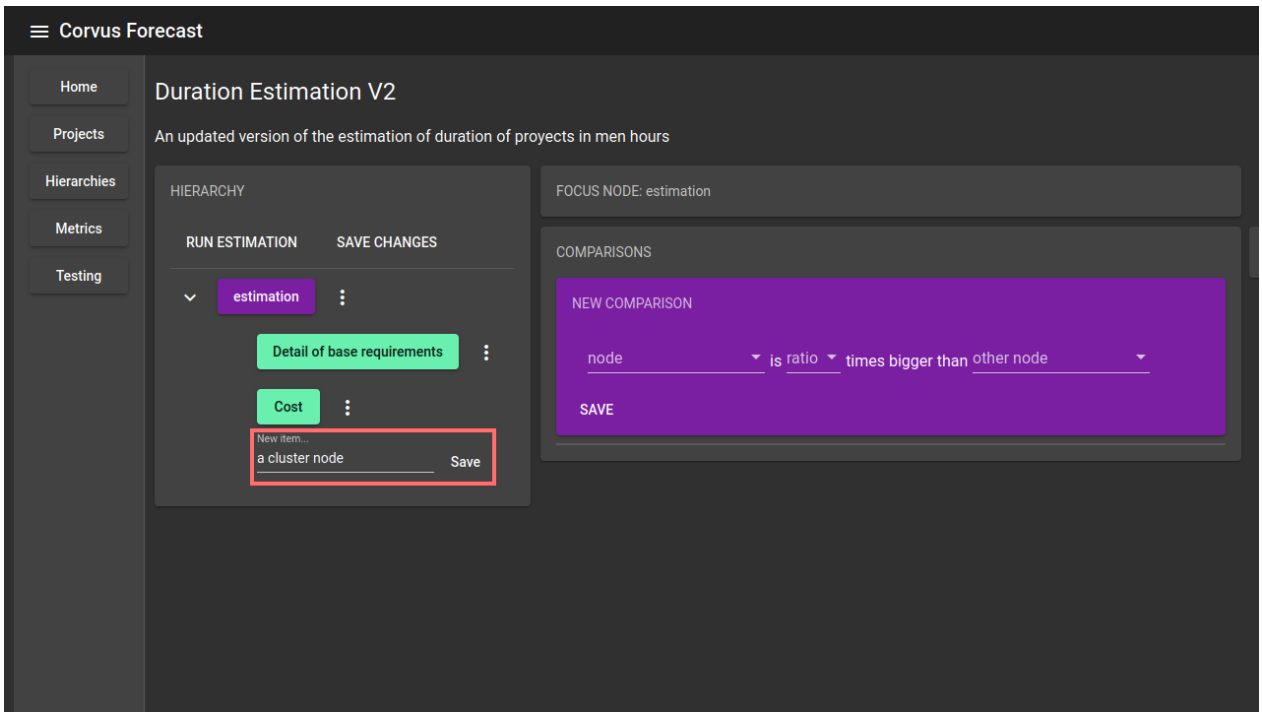
A continuación agregaremos un cluster a la jerarquía, para lo cual necesitamos un nodo jerárquico. Para crear un nodo jerárquico, esta vez en el menú del nodo, escogemos la opción para los nodos jerárquicos.



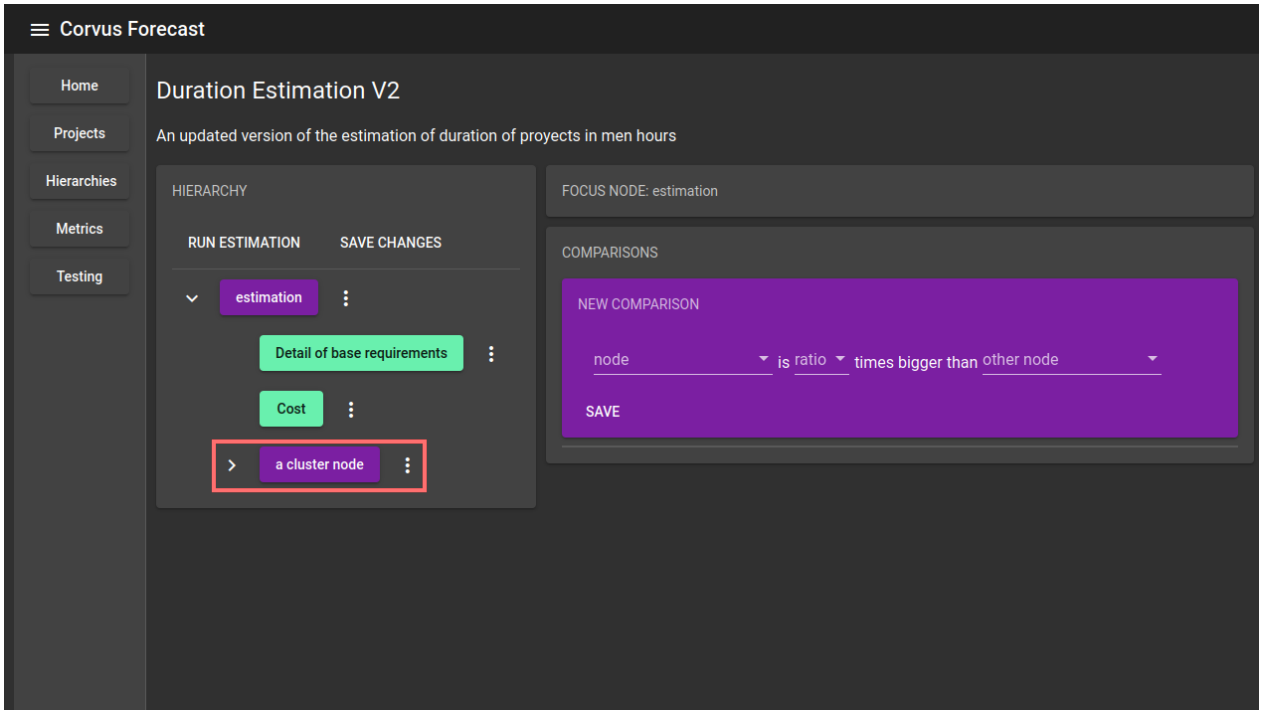
Y una nueva entrada es agregada a la jerarquía, esta vez, un cuadro de texto donde le damos un nombre al nodo.



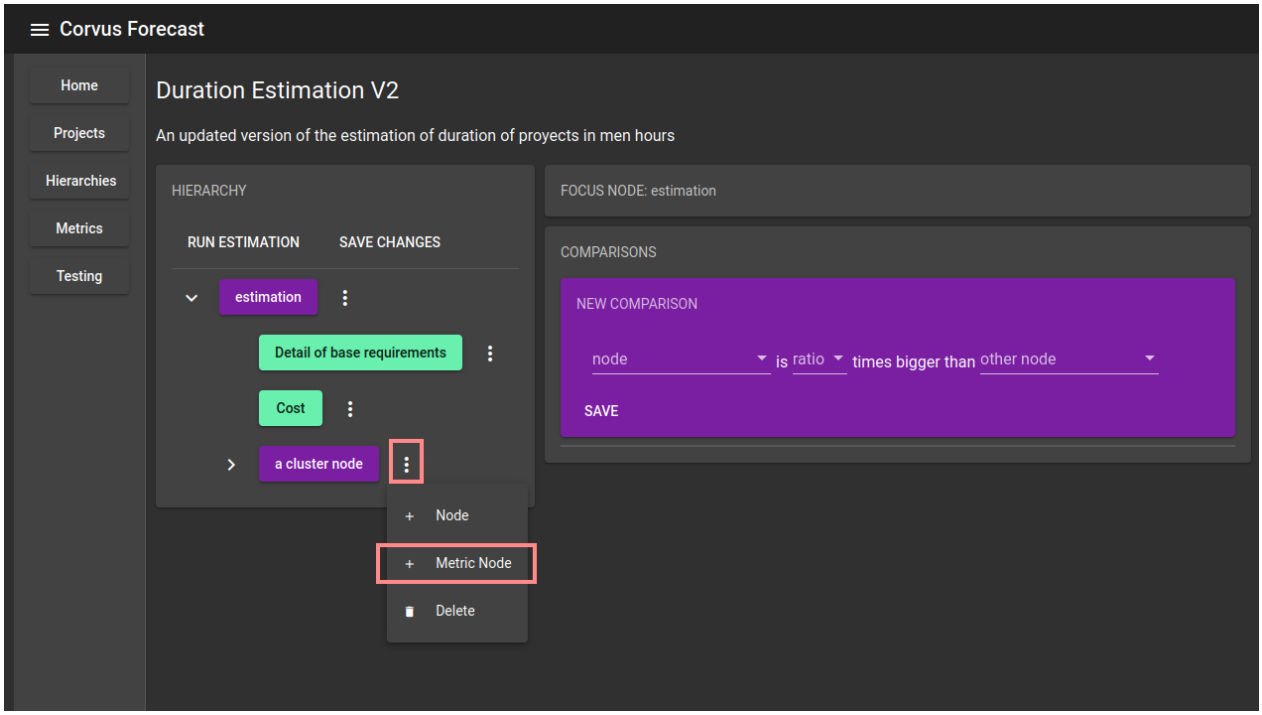
Escogemos un nombre descriptivo para el nodo y guardamos.



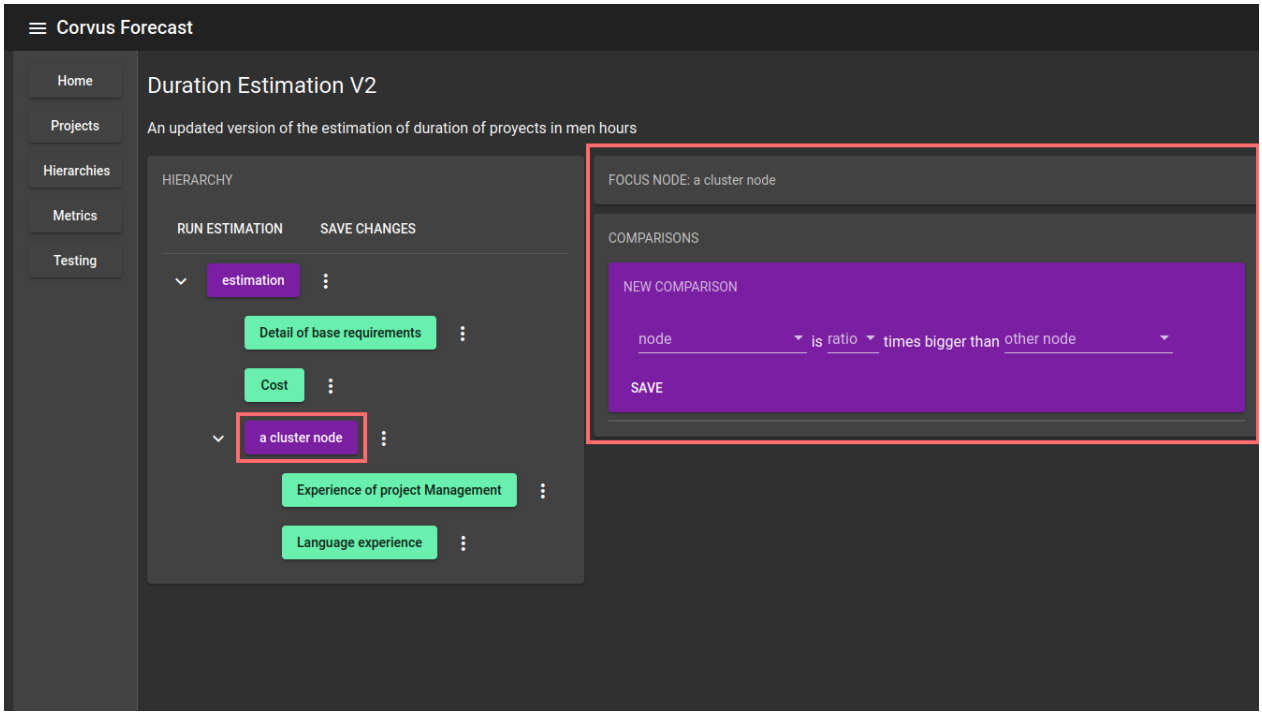
Creándose el nodo en la jerarquía.



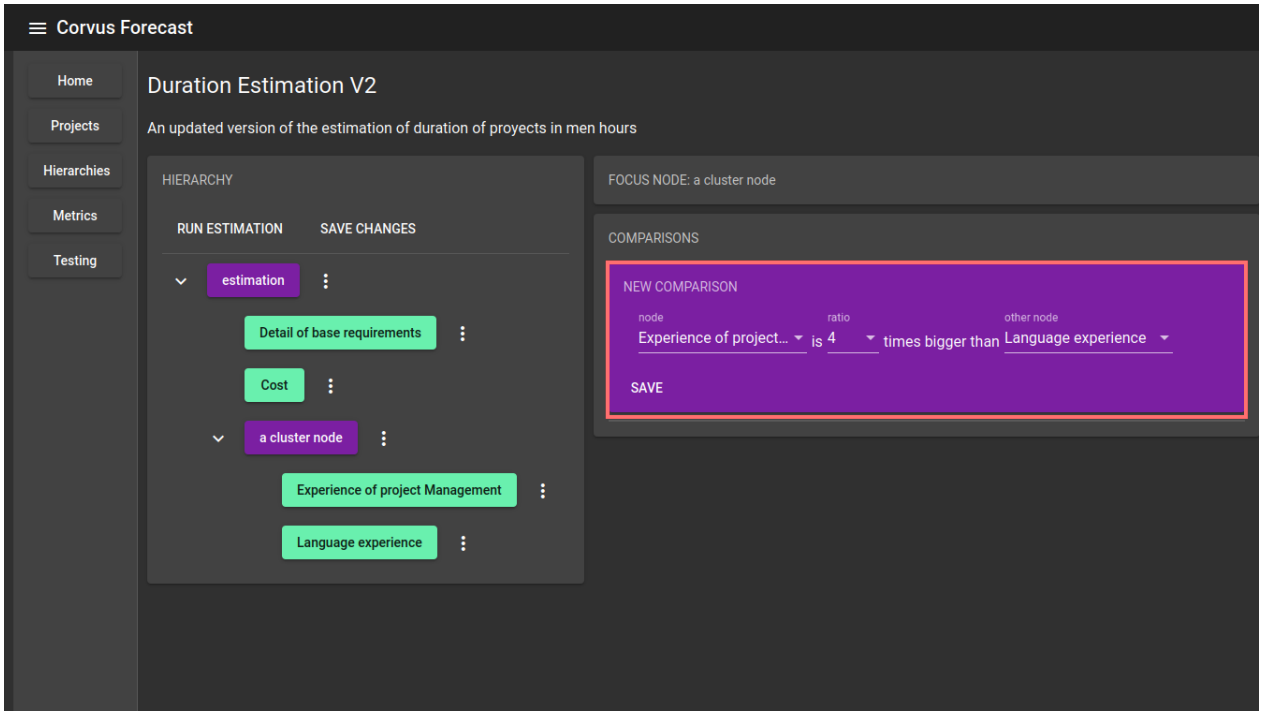
Por último, poblamos el nuevo nodo jerárquico con las métricas que nos faltan por agregar a la jerarquía. Para esto usamos el menú de creación del nodo cluster, y repetimos los pasos para crear nodos métricos.



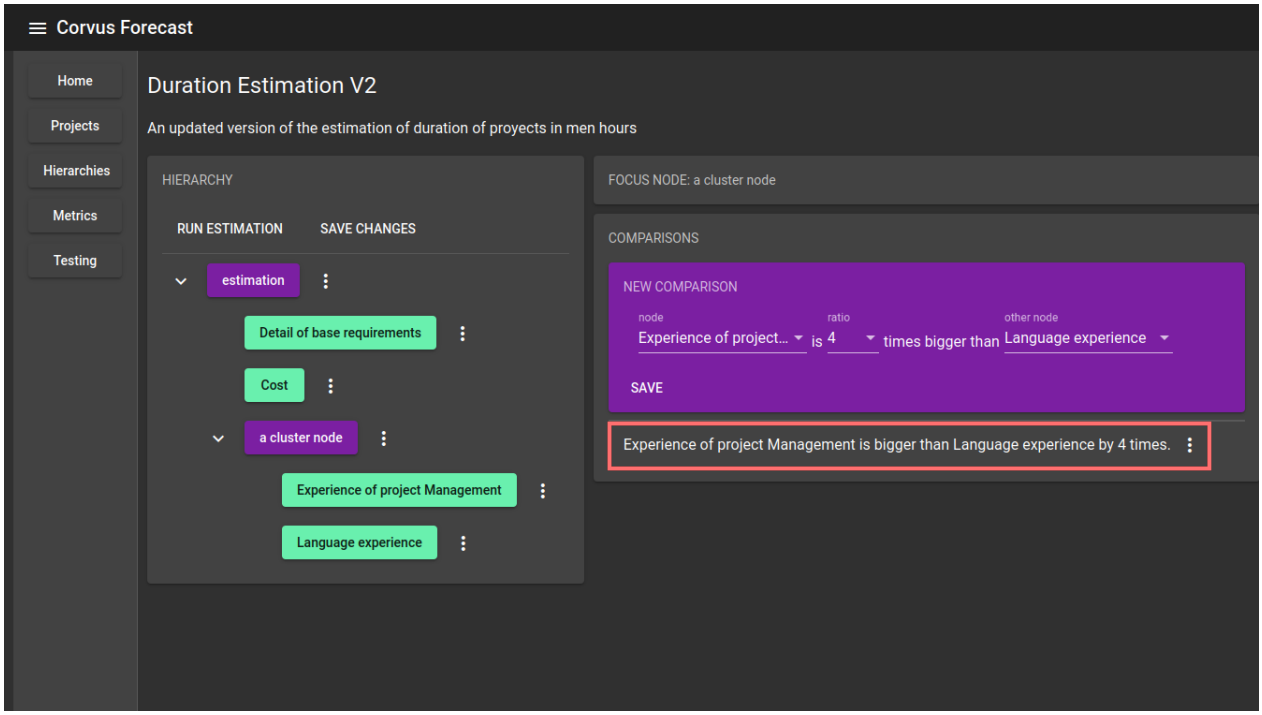
Con la jerarquía completa, procedemos a hacer las comparaciones relevantes a la jerarquía, las comparaciones en los nodos jerárquicos. Para esto seleccionamos el nodo que queremos modificar, el nodo cluster. La sección derecha de la interfaz cambia para mostrar la información del nodo seleccionado.



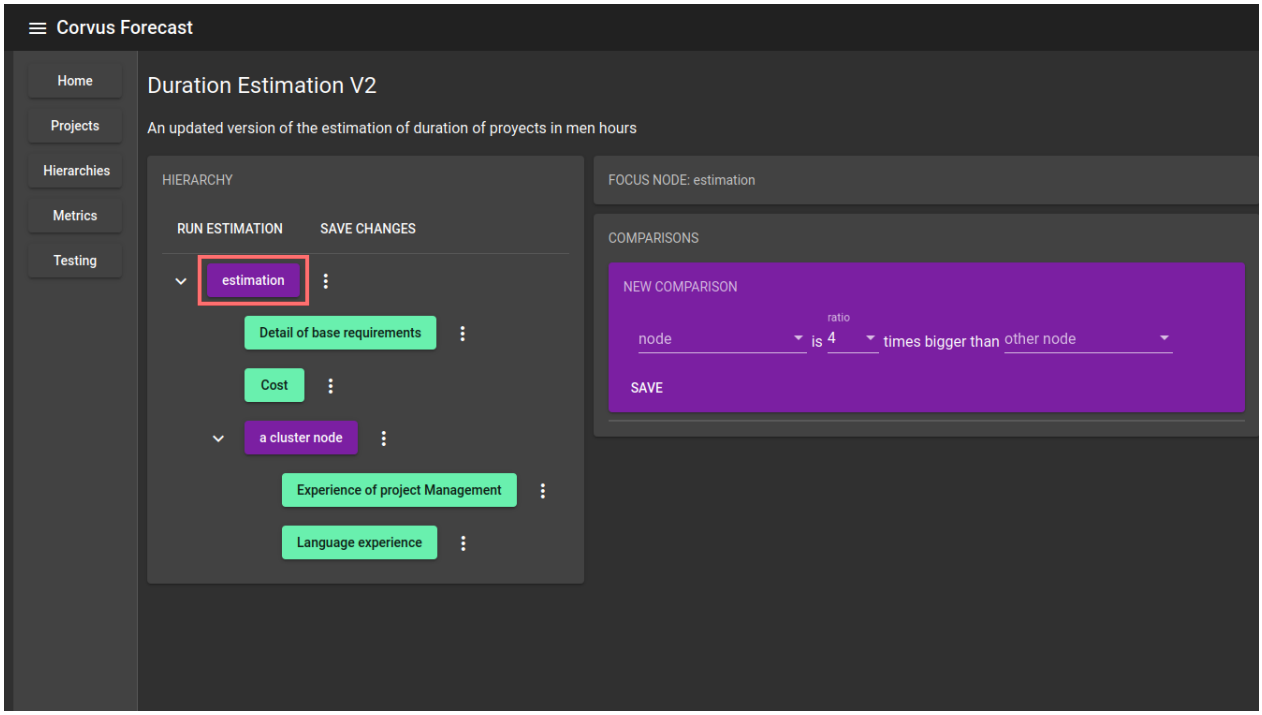
procedemos a comparar las métricas hijas de este nodo.



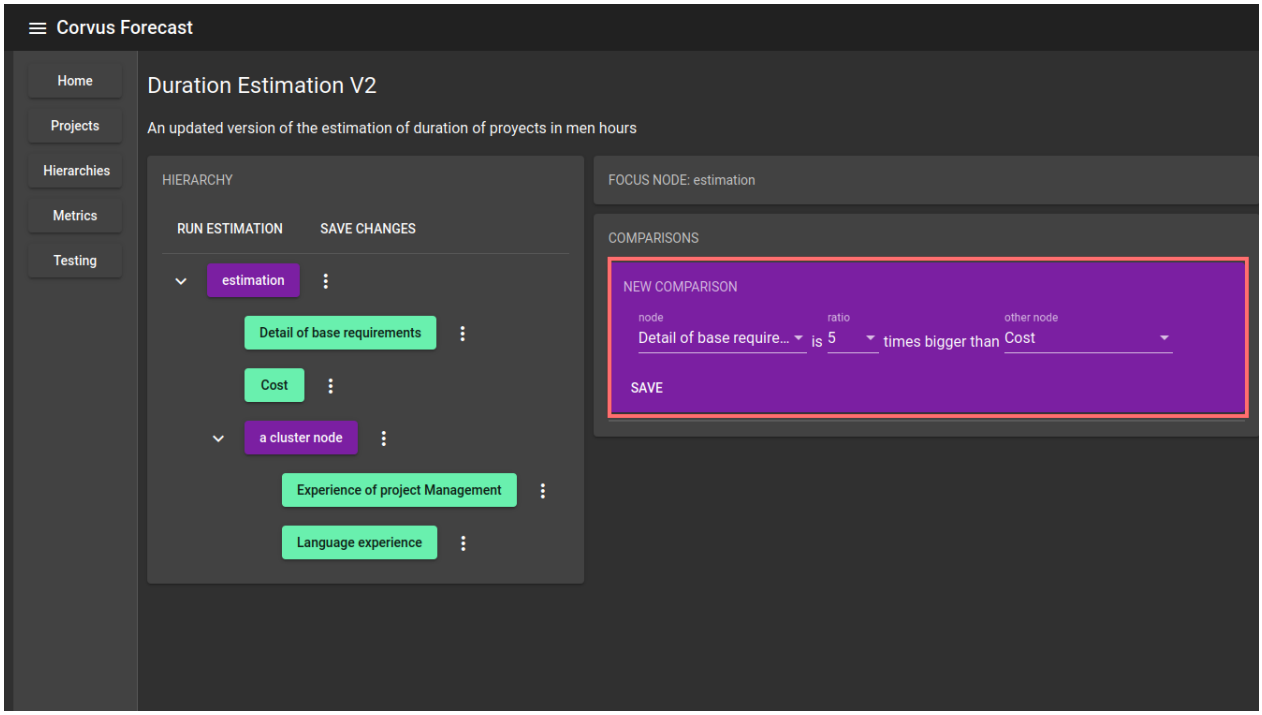
y la lista de comparaciones del nodo se actualiza, incluyendo esta nueva comparación.

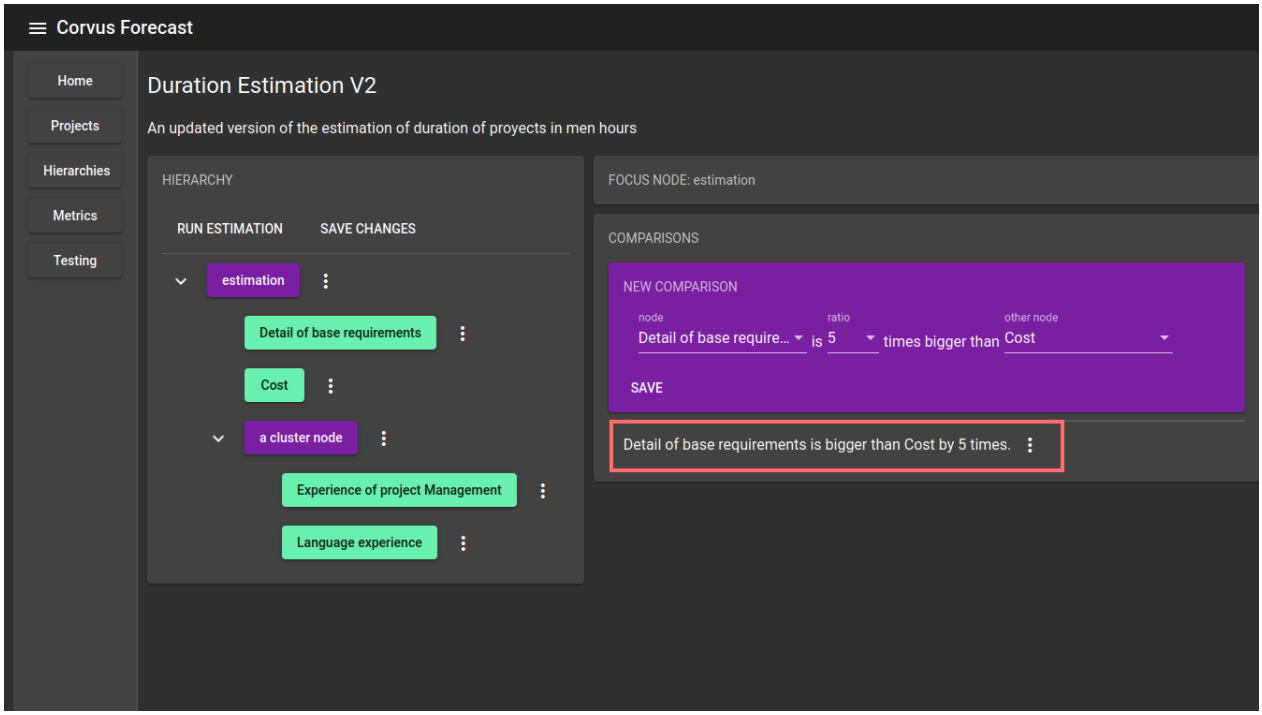


Escogemos ahora el nodo raíz de la jerarquía, el nodo de estimación, para comparar sus hijos.

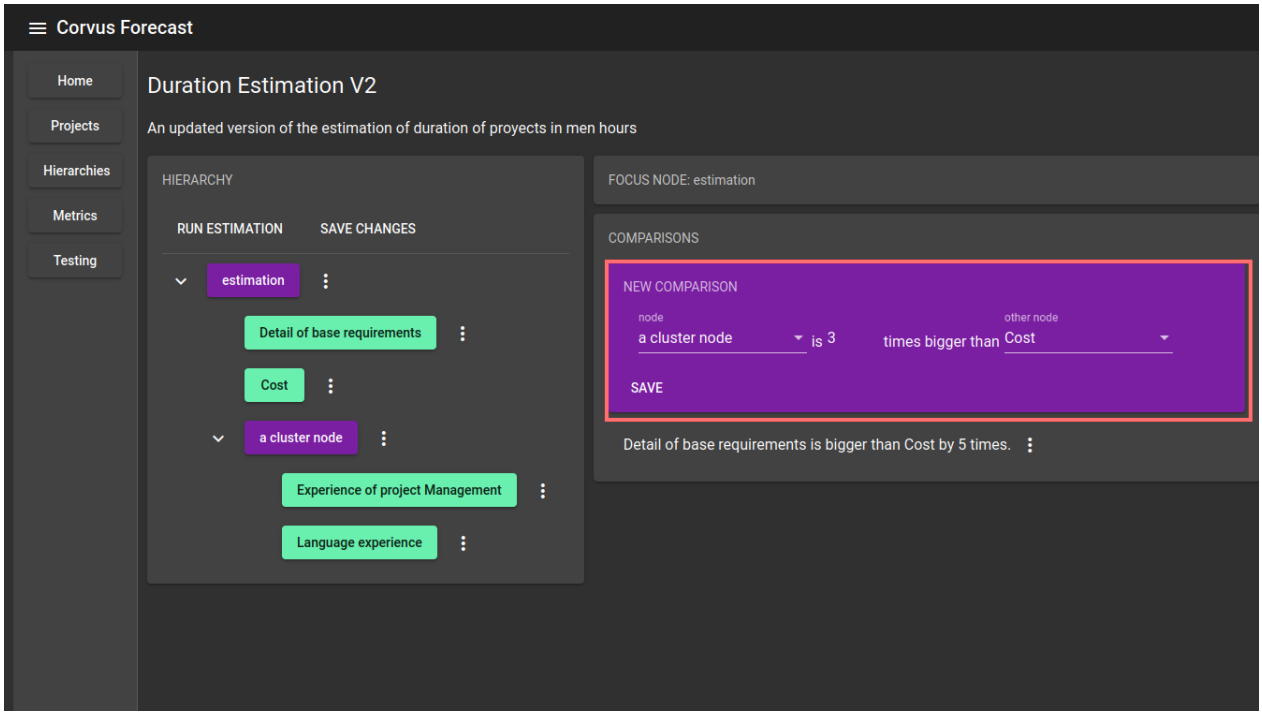


comparamos las métricas.

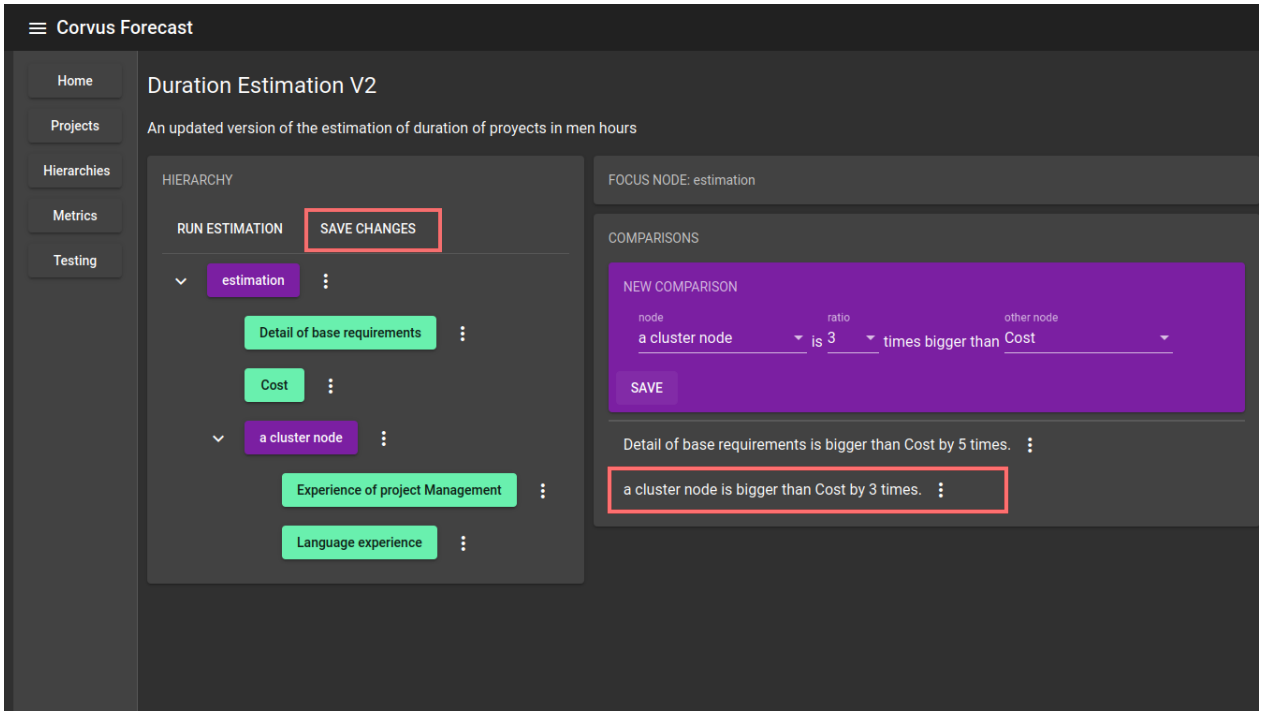




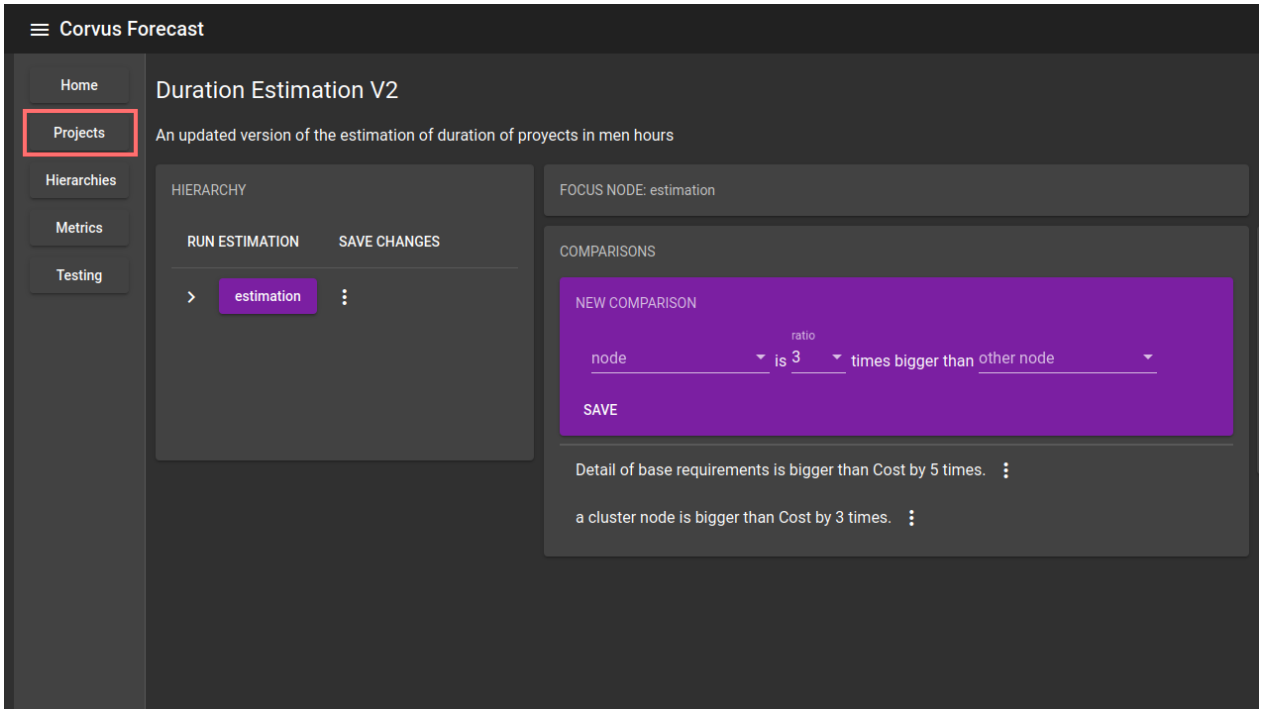
y comparamos el cluster con una métrica.



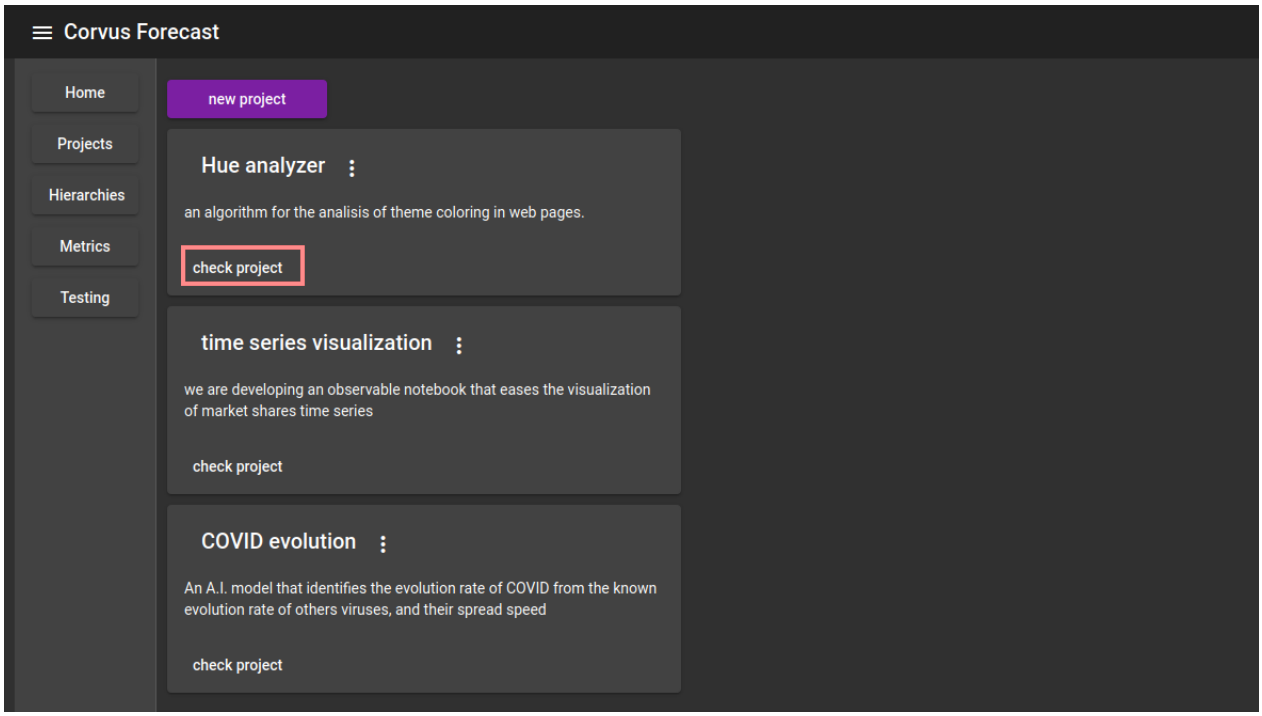
vemos la actualización de la lista de comparaciones, y terminamos las modificaciones de la jerarquía guardándola en la nube, usando el botón mostrado en la imagen.



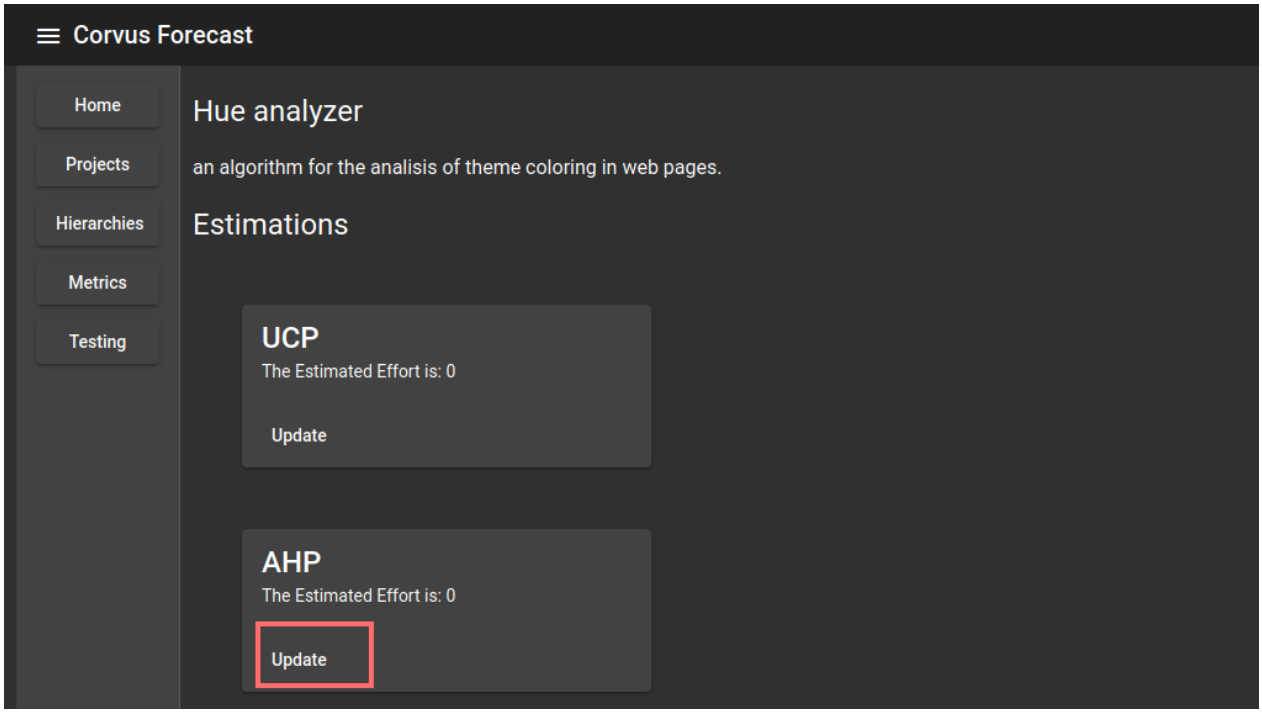
Ahora usaremos la nueva jerarquía para hacer la misma estimación que en el primer ejemplo. Para esto vamos a la página de proyectos.

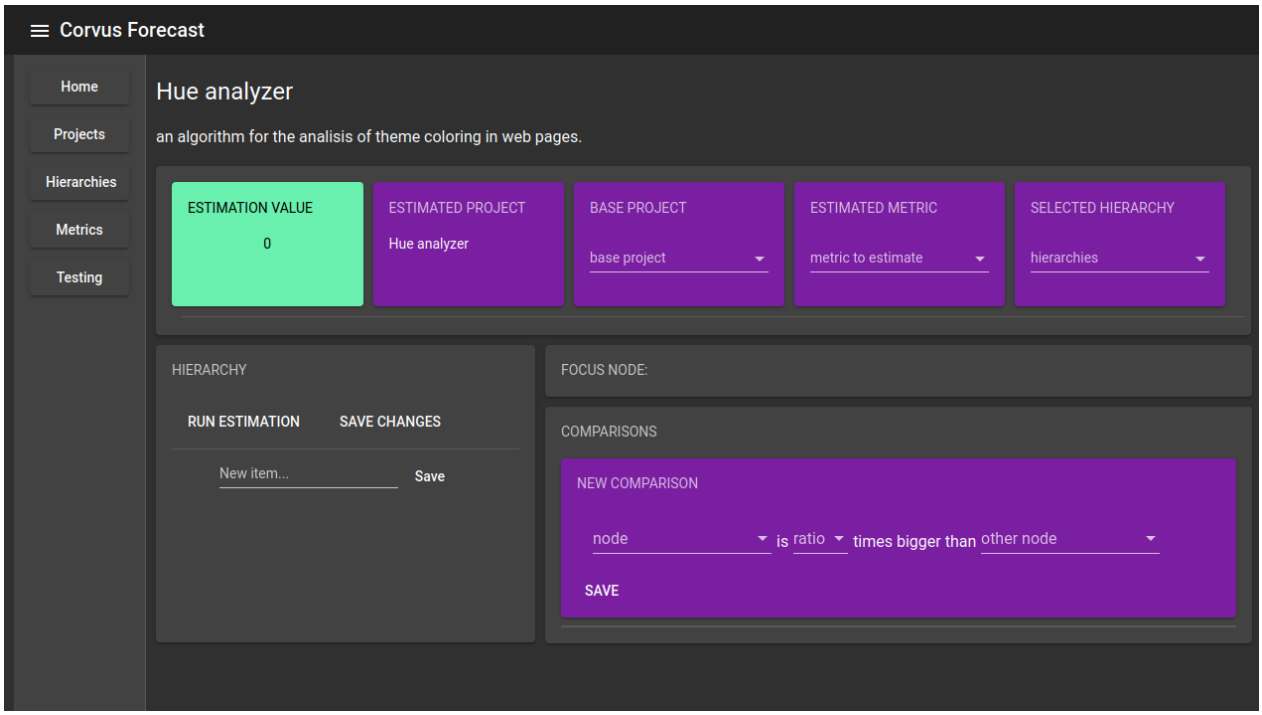


Y abrimos la página del proyecto.

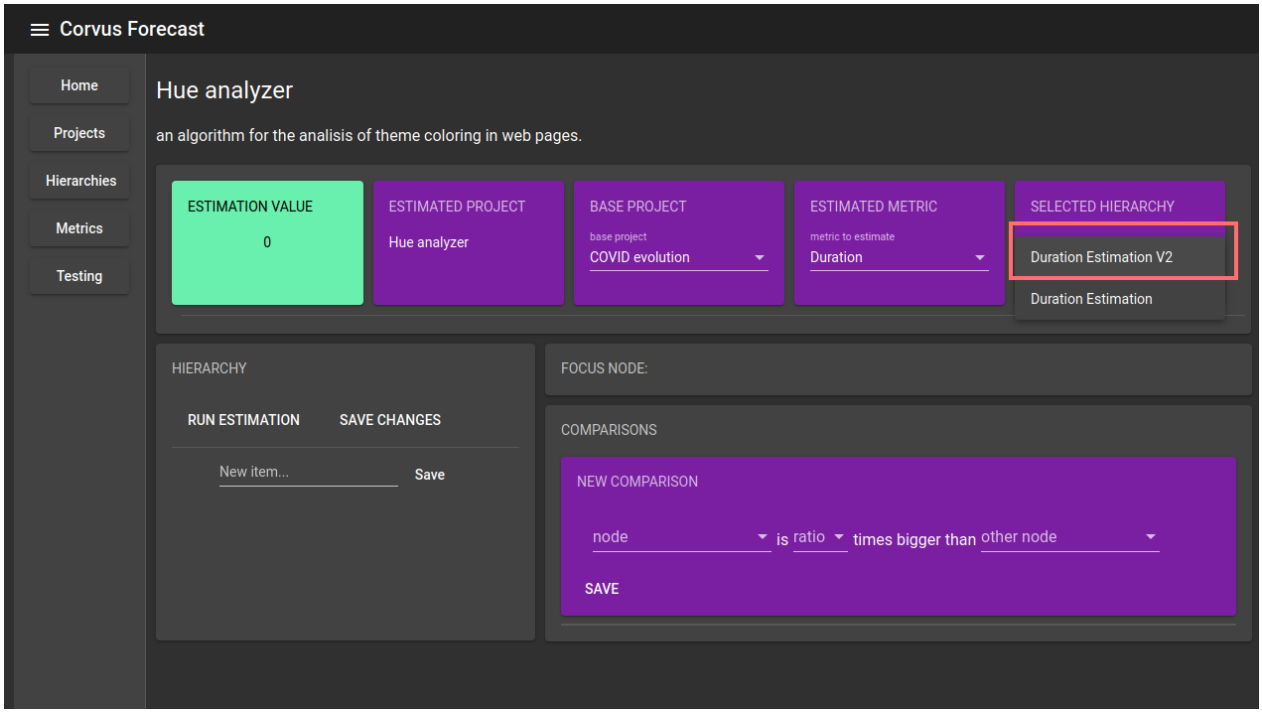


Seguimos con la página de la estimación del método AHP.

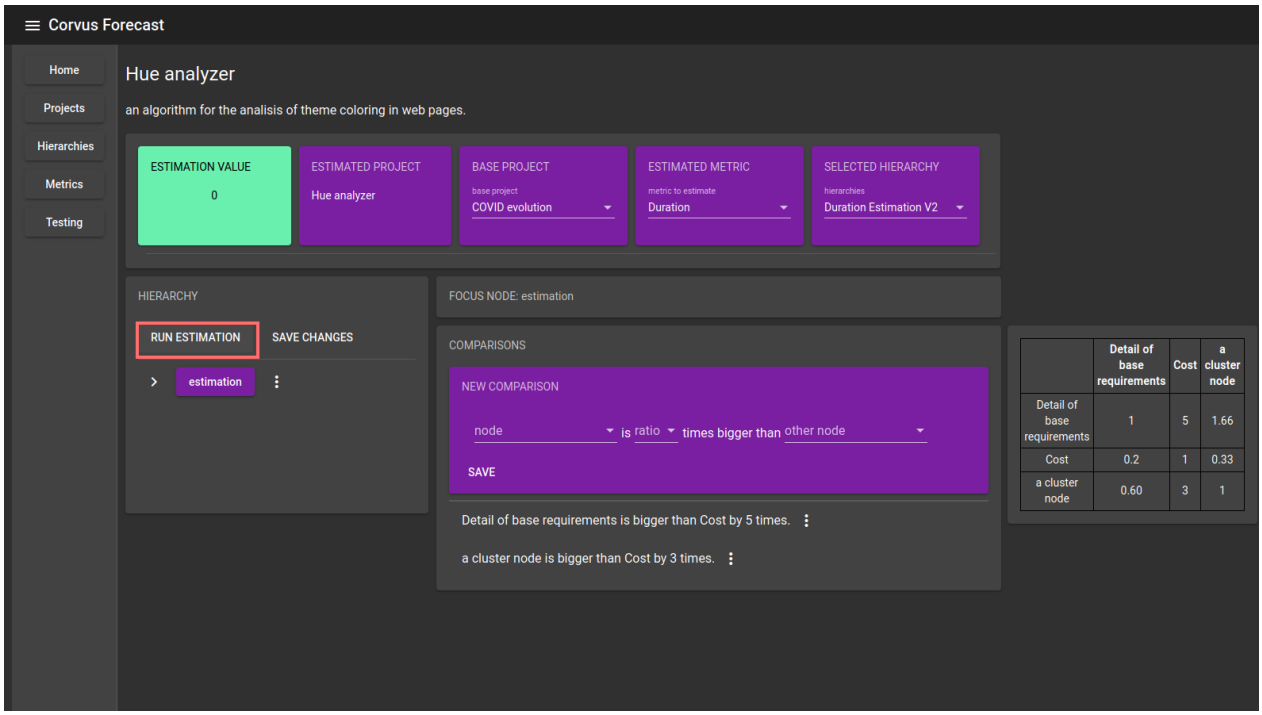




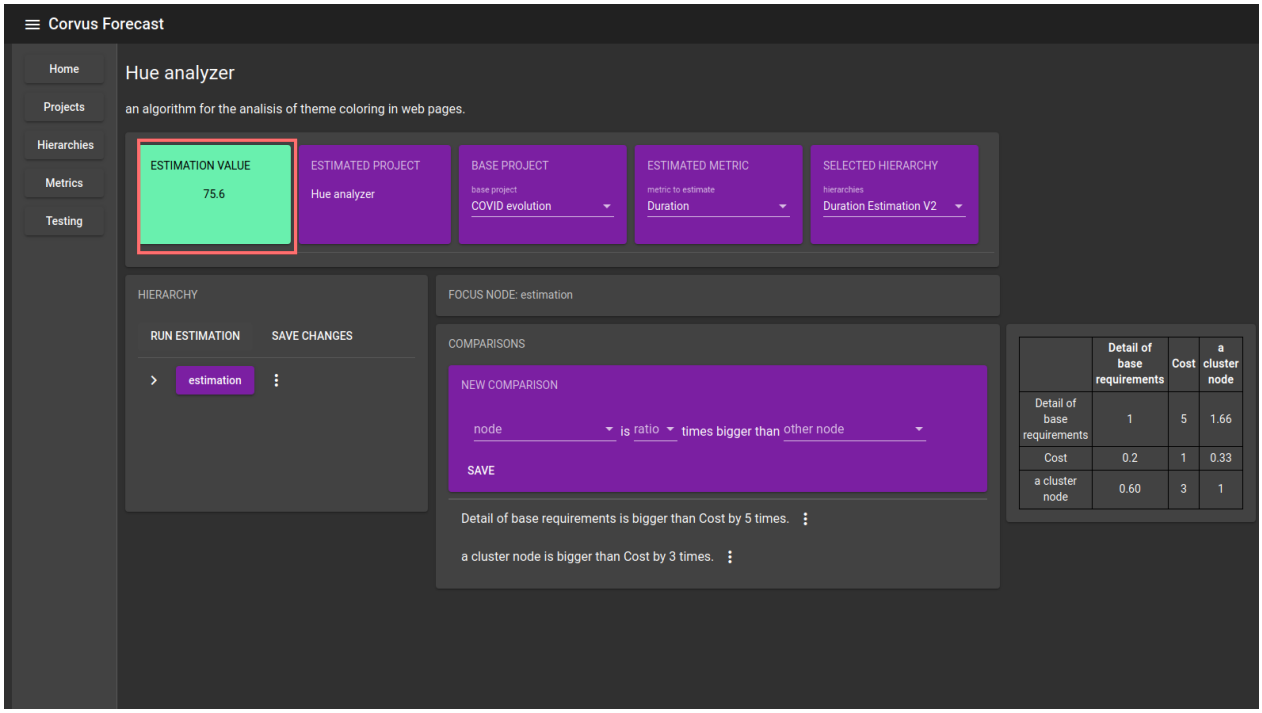
Escogemos el mismo proyecto base y la misma métrica a estimar que en el primer ejemplo. Esta vez escogemos la nueva jerarquía.



corremos la estimación .



y como todos los datos ya están el el sistema, al correr la estimación tenemos la estimación final sin ningún aviso.


Corvus Forecast
Home
Projects
Hierarchies
Metrics
Testing
Hue analyzer
an algorithm for the analisis of theme coloring in web pages.
ESTIMATION VALUE
75.6
ESTIMATED PROJECT
Hue analyzer
BASE PROJECT
base project
COVID evolution
ESTIMATED METRIC
metric to estimate
Duration
SELECTED HIERARCHY
hierarchies
Duration Estimation V2
HIERARCHY
RUN ESTIMATION
SAVE CHANGES
estimation
FOCUS NODE: estimation
COMPARISONS
NEW COMPARISON
node
is ratio
times bigger than
other node
SAVE
Detail of base requirements is bigger than Cost by 5 times.
a cluster node is bigger than Cost by 3 times.
Detail of base requirements
Cost
a cluster node
Detail of base requirements 1 5 1.66
Cost 0.2 1 0.33
a cluster node 0.60 3 1

7. Análisis Experimental

A continuación se exponen resultados de experimentos relevantes para el uso de la herramienta.

7.1. Mínimo error posible para las estimaciones

Para poder entender la calidad de las estimaciones de la técnica, veremos el mínimo error posible alcanzable en las estimaciones.

El error utilizado para los experimentos es la distancia porcentual de la estimación sobre el valor real. Así, si el valor real es 100 y la estimación es 110, el error es $\text{abs}(100-110)/100 * 100 = 10\,\%$ de error.

7.2. Jerarquía simple

Antes de utilizar una base de datos para las pruebas, veremos unos ejemplos simples de jerarquías, para entender mejor algunas de las limitaciones que tiene la técnica AHP.

Consideremos el siguiente contexto de proyectos y sus métricas:

Proyecto	Tiempo	Costo	Tamaño Equipo
A	1	1	3
B	2	2	2
C	3	3	3

En estos ejemplos estimaremos el tiempo a partir de las otras métricas. Armaremos tres jerarquías, una con el costo como única métrica, otra con el tamaño de equipo como única métrica, y la última con ambas métricas.

En todos los ejemplos que veremos, usamos el proyecto A como proyecto base, y estimamos el tiempo del proyecto B.

Partamos con la jerarquía que usa el costo.

Estimación Tiempo $\Big\{$ Costo

Manteniendo el orden A, B, C para los proyectos, las matriz de comparación para el costo es:

$$\begin{bmatrix} 1 & 1/2 & 1/3 \\ 2 & 1 & 2/3 \\ 3 & 3/2 & 1 \end{bmatrix}$$

Que resulta en la solución para el nodo de [1/6, 2/6, 3/6]

El nodo de la estimación de tiempo solo tiene un nodo hijo, el costo, por lo que la matriz

tiene un único elemento:

$$[1]$$

Que tiene la solución trivial [1].

Teniendo las soluciones para cada nodo de la jerarquía, lo que sigue es generar la solución para la jerarquía completa. El nodo de estimación de tiempo es trivial, por lo que la solución de la jerarquía es la misma que la solución del nodo de costo: [1/6, 2/6, 3/6].

Por último, para generar la estimación nos falta relacionar los tiempos y la solución de los proyectos en la jerarquía.

$$Tiempo_B = Estimacion_B$$
$$Tiempo_B = Tiempo_A * Sol_B/Sol_A$$
$$2 = 1 * \frac{2/6}{1/6}$$
$$2 = 2$$

A la izquierda tenemos el valor real para el tiempo del proyecto B, mientras que a la derecha tenemos la estimación del tiempo usando la jerarquía. En este ejemplo notamos dos cosas, que el valor real y la estimación toman el mismo valor, y que no hay comparaciones en esta jerarquía que puedan ser modificadas. La semejanza de la estimación con el valor real depende de la semejanza de la métrica en la jerarquía y la métrica estimada.

Veamos ahora el siguiente ejemplo, estimar el tiempo a partir del tamaño de equipo.

Estimación Tiempo $\Big\{$ Tamaño Equipo

Que manteniendo el orden de A, B, C, la matriz de comparación para el tamaño de equipo es:

$$\begin{bmatrix} 1 & 3/2 & 1 \\ 2/3 & 1 & 2/3 \\ 1 & 3/2 & 1 \end{bmatrix}$$

Con solución de nodo de [3/8, 2/8, 3/8]

Al igual que en la jerarquía anterior, el nodo de estimación solo tiene un elemento, y su solución es trivial [1], por lo que la solución de la jerarquía es la solución del nodo métrico: [3/8, 2/8, 3/8].

Esto nos deja con la relación:

$$Tiempo_B = Estimacion_B$$
$$Tiempo_B = Tiempo_A * Sol_B/Sol_A$$
$$2 = 1 * \frac{2/8}{3/8}$$
$$2 = 2/3$$

Esta vez, la estimación no es igual al valor real. El error porcentual es de abs$(2/3-2)/2 =$ 0,67, un 67 %.

Nuevamente, no hay partes de la jerarquía que el usuario tenga el poder de modificar para mejorar la estimación. La estimación es el resultado de la relación entre la métrica estimada y la métrica que forma parte de la jerarquía.

Por último, veremos una jerarquía que tiene dos métricas jerárquicas, el costo y el tamaño de equipo. Esta jerarquía contrasta con las anteriores ya que si tiene una comparación modificable por el usuario, como veremos.

La jerarquía que usamos es:

Estimación Tiempo $\begin{cases} \text{Costo} \\ \text{Tamaño Equipo} \end{cases}$

Laś soluciones de los nodos de costo y de tamaño de equipo son las mismas que en las jerarquías anteriores. La nueva matriz de esta jerarquía es la matriz no trivial del nodo de estimación de tiempo. El orden de los nodos hijos en este nodo es Costo, Tamaño Equipo. Y la matriz de comparaciones la podemos parametrizas como:

$$\begin{bmatrix} 1 & x \\ 1/x & 1 \end{bmatrix}$$

Con solución de [x/(x+1), 1/(x+1)], con x el valor de comparación que podemos cambiar.

Expresando ahora el resultado de la estimación a partir de este resultado obtenemos:

$$Tiempo_B = Estimacion_B$$
$$Tiempo_B = Tiempo_A * Sol_B/Sol_A$$
$$2 = 1 * (x * 2/6 + 1 * 2/8)/(x * 1/6 + 1 * 3/8)$$
$$2 = (16x + 12)/(8x + 18)$$

La cual es una solución en función del valor de la comparación x. Si graficamos esta función según la comparación x, podemos entender mejor su variación.

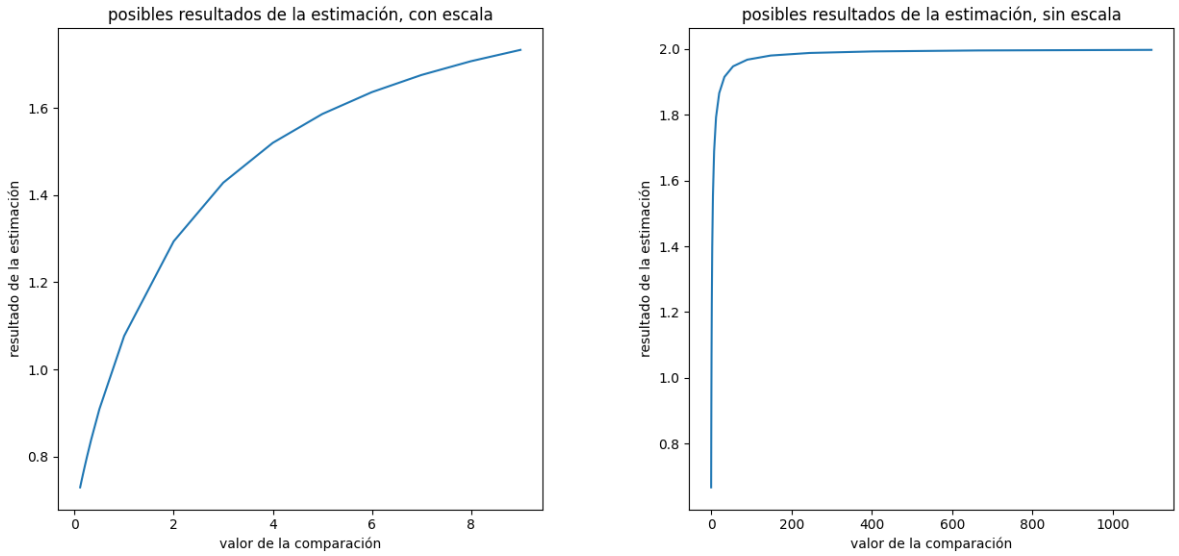


Figura 1: Estimación de la jerarquía según el valor de comparación, con y sin escala.

El gráfico nos muestra que al variar la comparación, conseguimos movernos entre el valor de ambas estimaciones, 2 y 2/3. Vemos también que al usar la escala, el rango de valores que puede tomar es reducido en comparación a la libertad de elección del valor de la comparación. Cabe recordar que la gracia de la escala es la facilidad para el uso humano.

7.3. Resolución automática de la jerarquía

A continuación se hacen pruebas usando una base de datos. Para correr las pruebas se implementó un resolvedor automático de jerarquías que, dada una jerarquía, un proyecto base y un proyecto a estimar con un valor real, ajusta las comparaciones en la jerarquía para lograr que la estimación de la jerarquía sea lo más cercano al valor real del proyecto estimado, computando así el mínimo error posible para las estimaciones.

El algoritmo utilizado para crear las jerarquías es un algoritmo genético sobre las comparaciones en la jerarquía. El algoritmo crea una generación de jerarquías que varían de forma aleatoria respecto a una jerarquía base. De estas jerarquías mantenemos la mejor como base para una nueva generación y repetimos.

La jerarquía que se crea para estas pruebas es una jerarquía plana, con un solo nodo jerárquico, el de estimación, y todos los nodos métricos como hijos de este.

Para mantener la semejanza con una jerarquía arbitraria, permitimos que las comparaciones tomen cualquier valor posible. El objetivo de la estructura jerárquica es facilitar las comparaciones entre métricas semejantes en tamaño. Si no tenemos restricción de tamaño, y las comparaciones las hace la computadora, entonces una jerarquía plana consigue lo mismo que una jerarquía más profunda.

7.4. Base de datos utilizada, Finnish Software Effort Dataset

La principal característica que se buscaba en un base de datos para ser utilizada era que estuviese disponible de manera libre, y que tuviese varias métricas numéricas ordenadas, además de una métrica de tiempo para hacer estimaciones típicas.

La búsqueda de tal base de datos se hizo primero en Google, para luego llevarse en el buscador especializado en bases de datos de Google (cuando fue descubierto).

La base de datos escogida que cumplió con estas condiciones es la Finnish Software Effort Dataset [4].

Varias métricas que presenta esta base de datos están numeradas del uno al veintiuno, y prefijadas con la letra T (T01, T02, ..., T21). Estas son las métricas que utilizamos en los experimentos para armar las jerarquías.

Los posibles valores que toman estas métricas son 1, 2, 3, 4 y 5. Y el significado de cada una se expresa en la siguiente lista (traducida de la página descriptiva de la base de datos).

- T01 = participación de los representativos del cliente
- T02 = rendimiento y disponibilidad del ambiente de desarrollo
- T03 = disponibilidad del personal de TI
- T04 = número de stakeholders
- T05 = presión del programa (schedule)
- T06 = impacto de los estándares
- T07 = impacto de los métodos
- T08 = impacto de las herramientas
- T09 = nivel de la administración de cambios
- T10 = madurez del proceso de desarrollo de software
- T11 = complejidad lógica del software
- T12 = Tamaño de la base de datos según el número de entradas
- T13 = número de interfaces a otros softwares
- T14 = requerimientos de calidad del software
- T15 = requerimientos de eficiencia del software
- T16 = requerimientos de instalación y entrenamiento del software
- T17 = habilidad de análisis del personal
- T18 = conocimiento relevante para la aplicación del personal
- T19 = habilidad del personal con las herramientas
- T20 = experiencia de la administración de proyectos
- T21 = habilidades de equipo del equipo del proyecto

La métrica que estimaremos en todas las pruebas es la duración de los proyectos. La duración de los proyectos en la base de datos está en meses.

La duración de un proyecto es una opción realística para hacer una estimación, mientras que las métricas prefijadas con T son las métricas que no tienen que ver con el cálculo de los puntos de función que también están presentes en la base de datos.

7.5. Variación de proyectos

Pasando ahora a las pruebas con el uso de la base de datos, queremos ver la influencia que tiene la inclusión de más proyectos a los datos utilizados para la estimación.

Para la jerarquía consideraremos las dos primeras métricas, T01 y T02, y veremos la variación del error porcentual según la cantidad de proyectos.

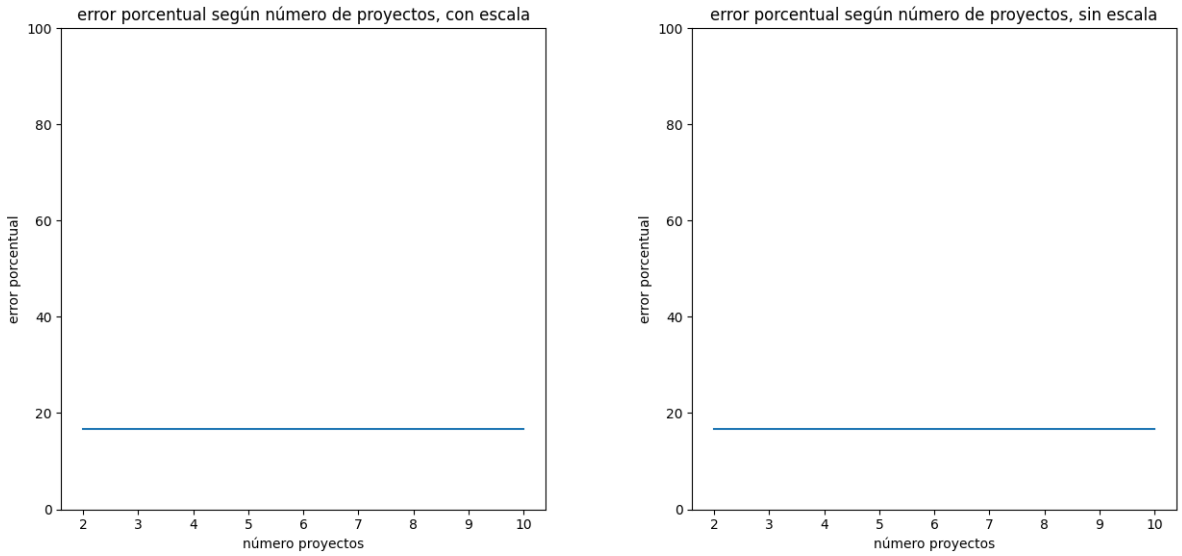


Figura 2: Error porcentual de la estimación según la cantidad de proyectos, con y sin escala.

Como se ve en los gráficos, la variación de la cantidad de proyectos no influye en el resultado de la estimación, ni tampoco influye el uso o el no uso de la escala.

Una posibilidad de este comportamiento es el uso arbitrario de las métricas T01 y T02 para la prueba, o la poca cantidad de proyectos utilizada para generar los gráficos, por lo que a continuación veremos varios casos. Cada caso considerado varía entre una y cincuenta proyectos, y los casos corresponden a jerarquías con 1, 2, 3, 5, 10 y 15 métricas (en orden, de T01 en adelante).

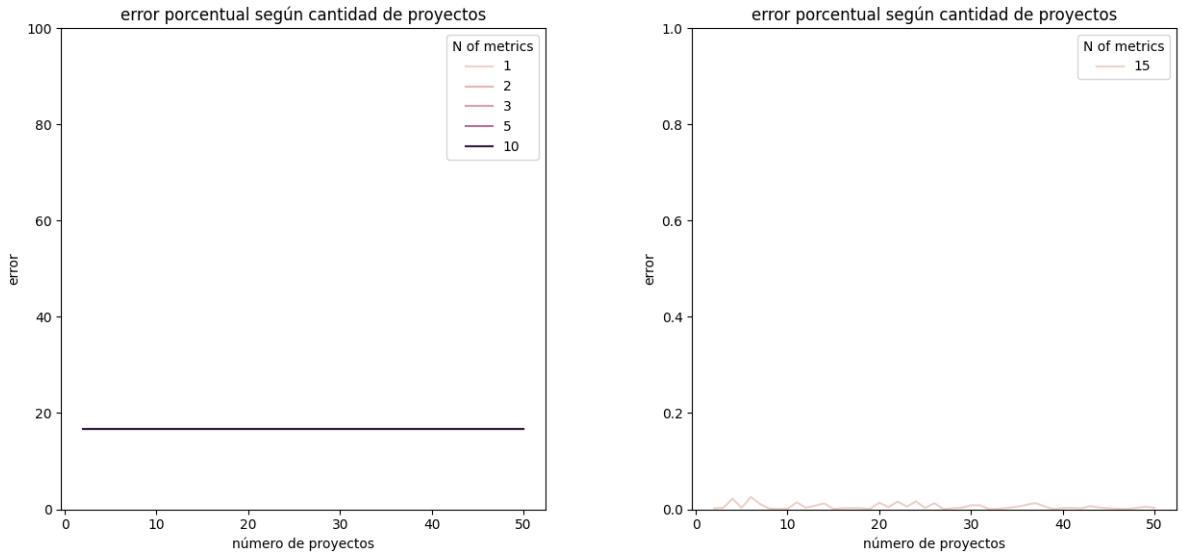


Figura 3: Error porcentual de la estimación según la cantidad de proyectos, para jerarquías de 1, 2, 3, 5, 10 y 15 métricas.

Como se ve en los gráficos, todos los casos menos el de 15 métricas se mantienen constantes en su error porcentual. El error para el caso con 15 métricas es menor al 0.2 %. Lo que nos lleva a que el error en la estimación no varía según el número de métricas utilizable, si no que depende de las métricas particulares utilizadas en cada caso. La escala de la imagen derecha va desde el 0 por ciento al 100 por ciento, mientras que la escala a la izquierda va desde el 0 por ciento al 1 por ciento

Las variaciones que se aprecian en la estimación de 15 métricas es causada por la generación automática de jerarquías, y no por la cantidad de proyectos.

7.6. Variación de métricas

El siguiente experimento consiste en variar la cantidad de métricas utilizadas, y analizar el error porcentual. Como la cantidad de proyectos no muestran una influencia sobre la estimación producida, consideramos únicamente dos proyectos para construir la jerarquía, el proyecto a estimar y el proyecto base. La métrica que estimamos sigue siendo la duración.

Lo primero que queremos ver es el error de estimación para las jerarquías con una sola métrica. Si lo discutido en la jerarquía simple de esta sección se mantiene correcto, entonces una de las primeras métricas por su cuenta debiese entregar una estimación cercana al 20 %, y una métrica cercana a la decimoquinta debiese entregar una estimación cercana al 0 %.

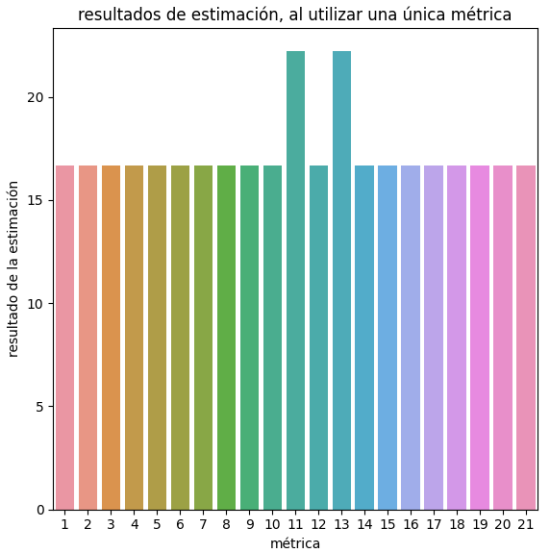


Figura 4: error porcentual de la estimación de una jerarquía para cada métrica

Lo que nos muestra el gráfico es que a partir de cada una de las métricas genera una estimación con un error cercano al 20 %. La mayoría genera un error del 16 % mientras que las estimaciones con T11 y T13 generan estimaciones con un error del 22 %.

A continuación veremos como se comporta una jerarquía al ir aumentando la cantidad de métricas, partimos con una jerarquía con las dos primeras (T01 y T02) hasta tener una jerarquía con todas las métricas (T01, ..., T21).

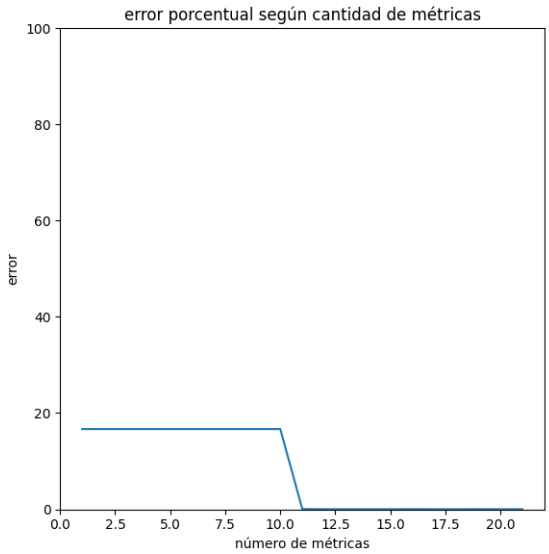


Figura 5: error porcentual de la estimación de una jerarquía según la cantidad de métricas utilizadas.

Al igual que en la variación de los proyectos, la estimación no mejora al tener una mayor cantidad de métricas, manteniéndose el error constante. Por lo que vemos en el gráfico, la estimación mejora al momento de incluir una métrica en particular, la métrica T11. Cuando esta se incluye, el error de estimación disminuye a 0 %.

Una jerarquía con la métrica T11 como única métrica tiene un error del 22 %, y una jerarquía con métricas T01 a la T10 tienen un error del 16 %, por lo que esperaríamos que una jerarquía que incluya ambas logre a lo menos un error del 16 %, como se explica en el ejemplo de una jerarquía simple en esta sección.

A continuación veremos la influencia que tienen las métricas sobre la estimación. Para esto armamos todas las jerarquías a partir de los pares de métricas que disponemos.

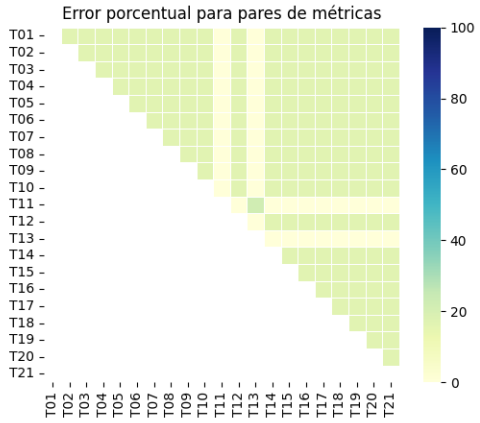


Figura 6: error porcentual de la estimación para jerarquías de cada par de métricas.

Lo que nos dice el gráfico es que todas las métricas logran el mismo nivel de error a excepción de las métricas T11 y T13. Hasta ahora, todas las pruebas han utilizado el primer proyecto de la base de datos como el proyecto base y el segundo proyecto de la base de datos como proyecto a estimar. Estas características de las métricas no son lo suficientemente representativas, por lo que nos interesa ver el promedio de los errores cuando cambiamos el proyecto estimado. Para la siguiente prueba, consideramos el promedio de los errores de las estimaciones para los primeros 100 proyectos de la base de datos, manteniendo una jerarquía para cada par de métricas.

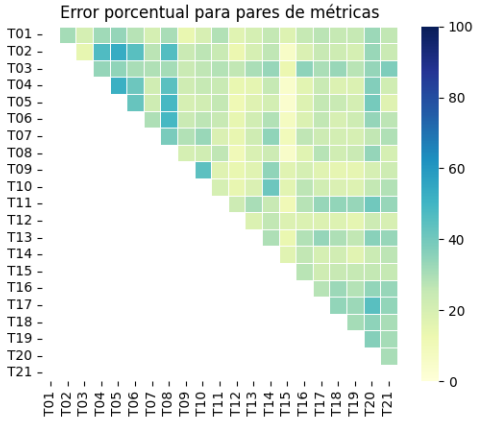


Figura 7: error promedio para las jerarquías de cada par de métricas.

Mostrando que en efecto la influencia que presentaban las métricas T10 y T12 eran situacionales.

7.7. Uso de la jerarquía automática en proyectos no ajustados

Por último, nos interesa saber la calidad de las jerarquías generadas automáticamente. Veremos qué tan capaces son de generar estimaciones para proyectos a los cuales no fueron ajustados. Mantenemos una estructura similar al experimento anterior. Volvemos a armar una jerarquía para cada par de métricas, pero esta vez calculamos el promedio de las jerarquías ajustadas para cada proyecto, y evaluadas en todos los demás proyectos.

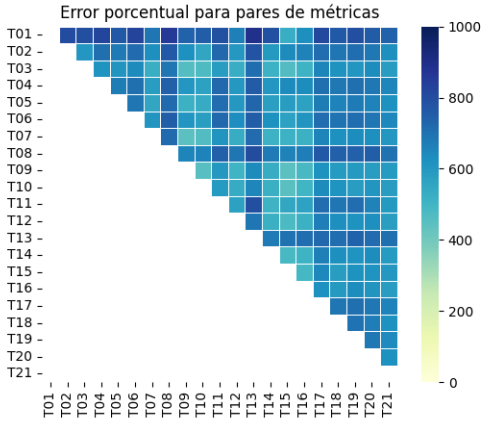


Figura 8: error promedio para las jerarquías evaluadas en otros proyectos, para cada par de métricas.

El gráfico anterior muestra la inutilidad de la jerarquía automática para armar jerarquías utilizables en la estimación de otros proyectos.

8. Discusión

8.1. Contexto ideal para el uso de la técnica

Siguiendo con la idea de una pequeña o mediana empresa, esta técnica se destaca por la poca cantidad de proyectos necesarios para generar una estimación. Basta tener otro proyecto con el cual comparar para comenzar a tener estimaciones.

Así mismo, no es necesaria una base estándar de métricas para utilizar la técnica, esta técnica es capaz de funcionar con un número variado de métricas, al no tener una jerarquía fija sino mutable.

8.2. Limitaciones de la técnica

El valor de mayor exactitud para las comparaciones es la comparación entre los valores reales de proyectos. Si se poseen estos valores reales, entonces tener más proyectos y sus comparaciones y valores no cambia la solución del proyecto estimado con valores reales, y en consecuencia no se obtiene una mejor estimación con mayor información (de otros proyectos).

Respecto a las métricas, tener un mayor número no aporta a la calidad de la estimación si estas métricas no tienen una mejor relación con la métrica estimada.

8.3. Mejoras a la técnica

Podemos modificar el último paso de la computación de la estimación, usando todos los proyectos presentes en todas las clausuras de la jerarquía en las que está el proyecto estimado, y no solo el proyecto base. Esto permitiría a usar todos estos proyectos como punto de referencia, y no un solo proyecto como punto de referencia.

Una vez que se dispone de valores reales de las métricas del proyecto base y del proyecto estimado, la información de los otros proyectos en estas métricas es ignorada. Encontrar una manera de incluir la información de más proyectos en las soluciones de las métricas jerárquicas ayudaría con la exactitud de la estimación.

8.4. Mejoras a la herramienta

En el análisis experimental se exploró el uso de un resolvedor automático de jerarquías. Su capacidad era reducida al generar jerarquías que solo estiman bien un solo proyecto. Se podría investigar la posibilidad de crear un generador automático de jerarquías que tome en consideración más proyectos, para obtener una jerarquía utilizable por el usuario para estimar proyectos cualquiera.

En la vista de estimación, la actualización de las comparaciones en una métrica es automática, así como las actualizaciones de los valores reales de proyectos. La modificación de la jerarquía y las comparaciones entre nodos en cambio no es automática y hay que guardarlas manualmente (con un botón), esto es porque se cierra la jerarquía visual cada vez que guarda. Se puede actualizar para no requerir del botón y que el guardado de la jerarquía sea automático.

En la misma vista, para producir una estimación, al momento de elegir una jerarquía, la herramienta requiere cargar todas las métricas, ya que luego de cargarlas las filtra para incluir la información de las métricas que forman parte de la jerarquía. Información que corresponde a las comparaciones de proyectos, y a valores reales de proyectos. Esta información es el grueso de los datos que mantiene la herramienta, que en caso de tener muchos proyectos con muchas métricas, la cantidad de información pedida al servidor es elevada. Por esto, la implementación de la herramienta no está pensada para usarse con varios proyectos o varias métricas, pues incurre en una elevada comunicación de datos, y en consecuencia, en un costo de mantención no ideal, y un mayor tiempo de respuesta al momento de cargar una jerarquía.

Las estimaciones generadas con esta técnica pueden ser utilizadas para generar comparaciones en jerarquías para otras estimaciones. La relevancia de las comparaciones producidas con estas estimaciones estarían por debajo de las estimaciones hechas por el usuario, que a su vez están por debajo de las comparaciones computadas a partir de los valores reales de los proyectos.

Las comparaciones computadas a partir de valores reales de métricas de proyectos son las más importantes al ser las más exactas. Estas comparaciones toman precedencia sobre las demás comparaciones. Esto implica que si se ingresan nuevos valores de métrica para un proyecto, estos pueden ocultar comparaciones hechas por el usuario. Una mejora consiste en informarle al usuario del cambio del uso de la comparación hecha por él a la comparación computada de los valores reales.

La herramienta está construida para soportar varios tipos de estimaciones y no solamente una estimación utilizando la técnica AHP simplificada. El uso de la técnica AHP simplificada requiere el tratamiento de las métricas aparte del tratamiento de las jerarquías, y aparte de estas dos el tratamiento de las estimaciones. Además, las estimaciones que se producen pueden ser para cualquier métrica, no solo una por proyecto como suelen ser hacer otras técnicas de estimación. La técnica AHP simplificada es más clara y más fácil de usar si no se mezcla con otras técnicas. Si la herramienta se dedica únicamente a la técnica AHP simplificada, dejando de lado el soporte para otras técnicas, su diseño podría reflejar de mejor manera el flujo de trabajo usado en las estimaciones con la técnica.

8.5. Otros

Esta herramienta permite el uso del AHP simplificado utilizando valores reales para las métricas de los proyectos, lo que convierte a esta técnica en una técnica híbrida, tanto algorítmica como analógica. Es una técnica analógica donde el usuario provee comparaciones entre proyectos y comparaciones entre métricas, y una técnica algorítmica, donde el usuario provee valores de proyectos para las métricas.

Una dificultad encontrada durante la creación de la herramienta fue la asociación de la jerarquía de nodos para la computación, con la estructura de árbol para la visualización. La asociación es un mapeo de uno a uno entre los nodos de la jerarquía con los nodos del árbol visual. Esto se convirtió en un problema en la etapa de creación de un nodo, ya que necesitábamos un nodo visual para mostrar la sección del ingreso de nombre del nodo. Pero para mostrar el nodo visual también requeríamos del nodo de computación perteneciente a la jerarquía. Tener este nodo sin nombre y sin información rompía el funcionamiento de la

herramienta. La solución tomada consistió en considerar para la computación únicamente nodos con nombre.

9. Conclusión

El objetivo de esta memoria se logró, la herramienta se construyó, y en efecto no se basa fuertemente en datos y está en un estado publicable. Esta memoria hizo un análisis de las técnicas de estimación existentes, análisis a partir del cual, junto a la caracterización del contexto de pequeñas y medianas empresas, se escogió una técnica a implementar.

Esta técnica electa, el AHP, fue simplificada para facilitar su uso y su adopción por los usuarios. Una vez la técnica estuvo clara, se implementó en una herramienta, y se prosiguió a su validación mediante el análisis de su comportamiento con datos reales de una base de datos de proyectos de software.

La técnica desarrollada, el AHP simplificado, es una técnica que mantiene características de la técnica original. mantiene la posibilidad del uso de comparaciones entre proyectos en vez del uso de datos reales.

Su principal característica es la flexibilidad. Flexibilidad en las métricas utilizadas, flexibilidad en la cantidad de métricas utilizadas y en la cantidad de proyectos utilizados.

Se implementó la reutilización de jerarquías, lo que disminuye el trabajo repetitivo necesario para hacer estimaciones por parte del usuario.

Se implementó la reutilización de información de proyectos, lo que también disminuye el trabajo repetitivo para hacer estimaciones por parte del usuario, pero el uso de más proyectos solo pasa a ser relevante cuando las comparaciones entre los proyectos son hechas a mano y no cuando son computadas a partir de valores reales. Se diseñó la herramienta con un enfoque en lo gráfico, como se aprecia en la sección de los casos de uso. Se diseñó la herramienta de manera extensible y se implementó una segunda técnica (UCP).

La técnica AHP simplificada tiene un mejor manejo si se trata sola en una herramienta, y no junto a otras técnicas. Esto debido a que por un lado trata las jerarquías, por otro las métricas y por otro los proyectos. Y esto sumado al hecho de que las estimaciones hechas con AHP puedan ser para cualquier métrica de un proyecto.

En contraste, las estimaciones de las demás técnicas son de una sola métrica como tamaño o tiempo, lo que resulta en una discordancia de resultados entre esta técnica y las demás.

Los análisis hechos de la técnica AHP simplificada muestran que se puede llegar a tener una estimación con un error menor al 20 %, considerado un valor aceptable para una técnica de estimación, pero estos mismos análisis nos llevan a creer que esto es resultado de la relación inherente entre métricas y no un resultado que se logre gracias a la técnica.

Bibliografía

[1] *awesome estimator*. https://estimator.cl. Accessed: 2021-07-28.

[2] Shilpa Bhalerao y Maya Ingle. "Incorporating Vital Factors in Agile Estimation through Algorithmic Method." En: *IJCSA* 6.1 (2009), págs. 85-97.

[3] Iran Boroujen. "A case study research on software cost estimation using experts' estimates, wideband delphi, and planning poker technique". En: *International Journal of Software Engineering and its applications* 8.11 (2014), págs. 173-182.

[4] Pekka Forselius Boyce Sigweni Martin Shepperd. *Finnish Software Effort Dataset*. http://doi.org/10.6084/m9.figshare.1334271.v3. 2016.

[5] Roy K Clemmons. "Project estimation with use case points". En: *The Journal of Defense Software Engineering* 19.2 (2006), págs. 18-22.

[6] *estimatool*. https://estimatool.com. Accessed: 2021-07-28.

[7] Standish Group y col. *The CHAOS report. 2015*. 2018.

[8] Philip M Johnson y col. "Empirically guided software effort guesstimation". En: *IEEE Software* 17.6 (2000), págs. 51-56.

[9] Jacky Keung. "Software development cost estimation using analogy: a review". En: *2009 Australian Software Engineering Conference*. IEEE. 2009, págs. 327-336.

[10] Izzat M Alsmadi y Maryam S Nuser. "Evaluation of cost estimation metrics: Towards a unified terminology". En: *Journal of computing and information technology* 21.1 (2013), págs. 23-34.

[11] Eduardo Miranda. "An evaluation of the paired comparisons method for software sizing". En: *Proceedings of the 22nd international conference on Software engineering*. 2000, págs. 597-604.

[12] Samson Wanjala Munialo y Geoffrey Muchiri Muketha. "A review of agile software effort estimation methods". En: (2016).

[13] Nolberto Munier, Eloy Hontoria y col. "Uses and Limitations of the AHP Method". En: *Management for Professionals* (2021).

[14] Naoki Ohsugi y col. "Effort estimation based on collaborative filtering". En: *International Conference on Product Focused Software Process Improvement*. Springer. 2004, págs. 274-286.

[15] *orange estimator*. https://sourceforge.net/projects/effort-estimate/. Accessed: 2021-07-28.

[16] Alejandro Pardo Rodriguez. "Un mecanismo de administración de información histórica de proyectos de software que facilite la estimación de los desarrollos". En: (2020).

[17] Thomas L Saaty y Luis G Vargas. "Estimating technological coefficients by the analytic hierarchy process". En: *Socio-Economic Planning Sciences* 13.6 (1979), págs. 333-336.

[18] Christopher Schofield. "An empirical investigation into software effort estimation by analogy". Tesis doct. Bournemouth University, 1998.

[19] Christian Seybold, M Glinz y Arun Mukhija. "Estimation Tools". En: (2003).

[20] Martin Shepperd y Michelle Cartwright. "Predicting with sparse data". En: *IEEE Transactions on Software Engineering* 27.11 (2001), págs. 987-998.

[21] Shadab Siddiqui, Mohd Rizwan Beg y Shahin Fatima. "Effectiveness of requirement prioritization using analytical hierarchy process (AHP) and planning game (PG): A comparative study". En: *PLANNING* 90 (2013), pág. 100.

[22] Louis M. Taff, James W. Borchering y RW Hudgins Jr. "Estimeetings: development estimates and a front-end process for a large project". En: *IEEE Transactions on software engineering* 17.8 (1991), pág. 839.

[23] Robert C Tausworthe. "The work breakdown structure in software project management". En: *Journal of Systems and Software* 1 (1979), págs. 181-186.

[24] Tomás Vera, Sergio F Ochoa y Daniel Perovich. "Survey of software development effort estimation taxonomies". En: *Computer Science Department, University of Chile: Santiago, Chile* (2017).