

# Modelação e Visualização Tridimensional em Arquitectura



UNIVERSIDADE  
DE LISBOA



FACULDADE DE ARQUITETURA  
UNIVERSIDADE DE LISBOA

Mestrado Integrado em Arquitectura  
Ano Lectivo 2022-2023 2º Semestre  
Docente - Nuno Alão 3º Ano

20191188



MAFALDA DE MELO COIMBRA AMARAL ALBANO

**U** LISBOA

UNIVERSIDADE  
DE LISBOA



FACULDADE DE ARQUITETURA  
UNIVERSIDADE DE LISBOA

**MVTA**

Mestrado Integrado em Arquitectura  
Ano Lectivo 2022-2023 2º Semestre  
Docente - Nuno Alão 3º Ano

# ÍNDICE

## **Exercício 1 – Superfície Parabólica**

## **Exercício 2 – Sólidos Platónicos**

### **Exercício 2.1**

*Exercício 2.1.1 – Tetraedro*

*Exercício 2.1.2 – Hexaedro*

*Exercício 2.1.3 – Octaedro*

*Exercício 2.1.4 – Dodecaedro*

*Exercício 2.1.5 – Icosaedro*

### **Exercício 2.2 – Relações de Dualidade**

*Exercício 2.2.1 – Tetraedro*

*Exercício 2.2.2 – Hexaedro | Octaedro*

*Exercício 2.2.3 – Dodecaedro | Icosaedro*

## **Exercício 3 – Cone**

*Exercício 3.1 – Interseção de Planos Secantes*

*Exercício 3.2 – Superfícies Resultantes*

## **Exercício 4 – Xadrez**

## **Exercício 5 – Hiperboloide e Comandos EDGESURF e LOFT**

*Exercício 5.1 – Hiperboloide de Revolução*

*Exercício 5.2 – Comandos EDGESURF e LOFT*

## **Exercício 6 – Museu Guggenheim**

## **Exercício 7 – Lamparina**

*Exercício 7.1 – Materiais e Texturas*

**Parábola** – superfície curva, cujos pontos se encontram à mesma distância de um ponto fixo (*Foco*) e de uma reta perpendicular ao eixo da sua projeção horizontal (*diretriz*).

Tendo estabelecido um eixo (*e*) e a diretriz da parábola, define-se, algures no eixo, o *Foco* da parábola. A meio da distância do *Foco* ao ponto de interseção do eixo com a *diretriz*, marca-se o vértice. Este será o ponto da parábola mais próximo da *diretriz* e, a partir do qual, a parábola se desenvolve.

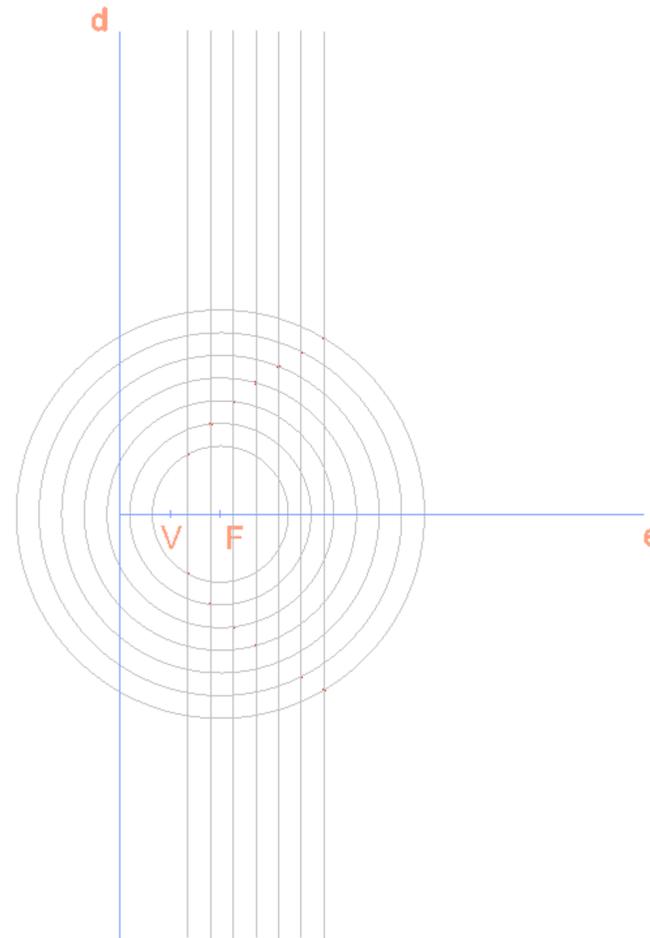
Os pontos da parábola definem-se na interseção de circunferências com centro no *Foco*, com retas paralelas à *diretriz* – para se garantir a equidistância dos pontos da parábola, simultaneamente, ao *Foco* e à *diretriz*, a circunferência inicial (que dará origem às restantes) deve distar exatamente o mesmo do *Foco*, que a primeira reta traçada, paralela à *diretriz* (o raio da circunferência deve ser maior que a distância do *Foco* ao vértice, e menor que a distância do *Foco* ao ponto de interseção da *diretriz* com o eixo).

Nas interseções da reta com a circunferência, marcam-se os dois primeiros pontos. Os restantes pontos da parábola surgem da marcação de novas circunferências (o comando **OFFSET** permite criar circunferências concêntricas, sempre com o mesmo distanciamento entre elas) e de novas retas paralelas à *diretriz* (a distância entre estas retas é definida pelo distanciamento definido para as circunferências).

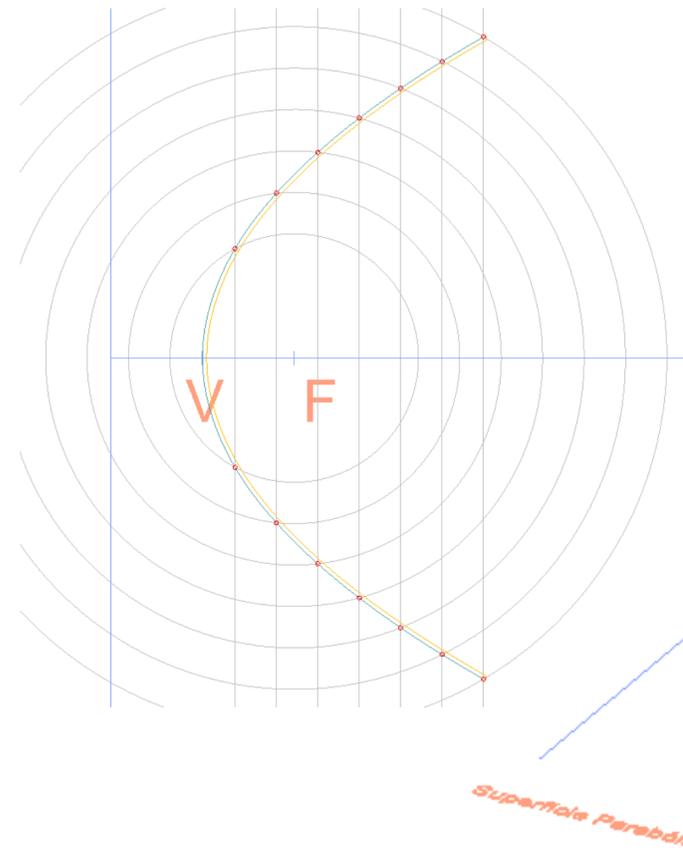
Quando se atinge a parábola pretendida, traça-se uma paralela a esta (para maior precisão do processo deve utilizar-se o comando **PEDIT**, de forma a criar-se uma **POLYLINE** - as parábolas serão fechadas com um semicírculo (unido à parábola de dentro – através do comando **JOIN**), que garantirá o fechamento das superfícies parabólicas.

A revolução da parábola a superfície parabólica é conseguida através do comando **REVSURF**. Determina-se um eixo de revolução (neste caso, *e* ou *d*) e um ângulo de revolução (180° e 360°, respetivamente, a superfície parabólica em baixo e a superfície parabólica em cima). Com o comando **SURFTAB** define-se a densidade da malha que forma a superfície tridimensional e com o comando **SHADE**, transforma-se esta mesma malha numa superfície opaca.

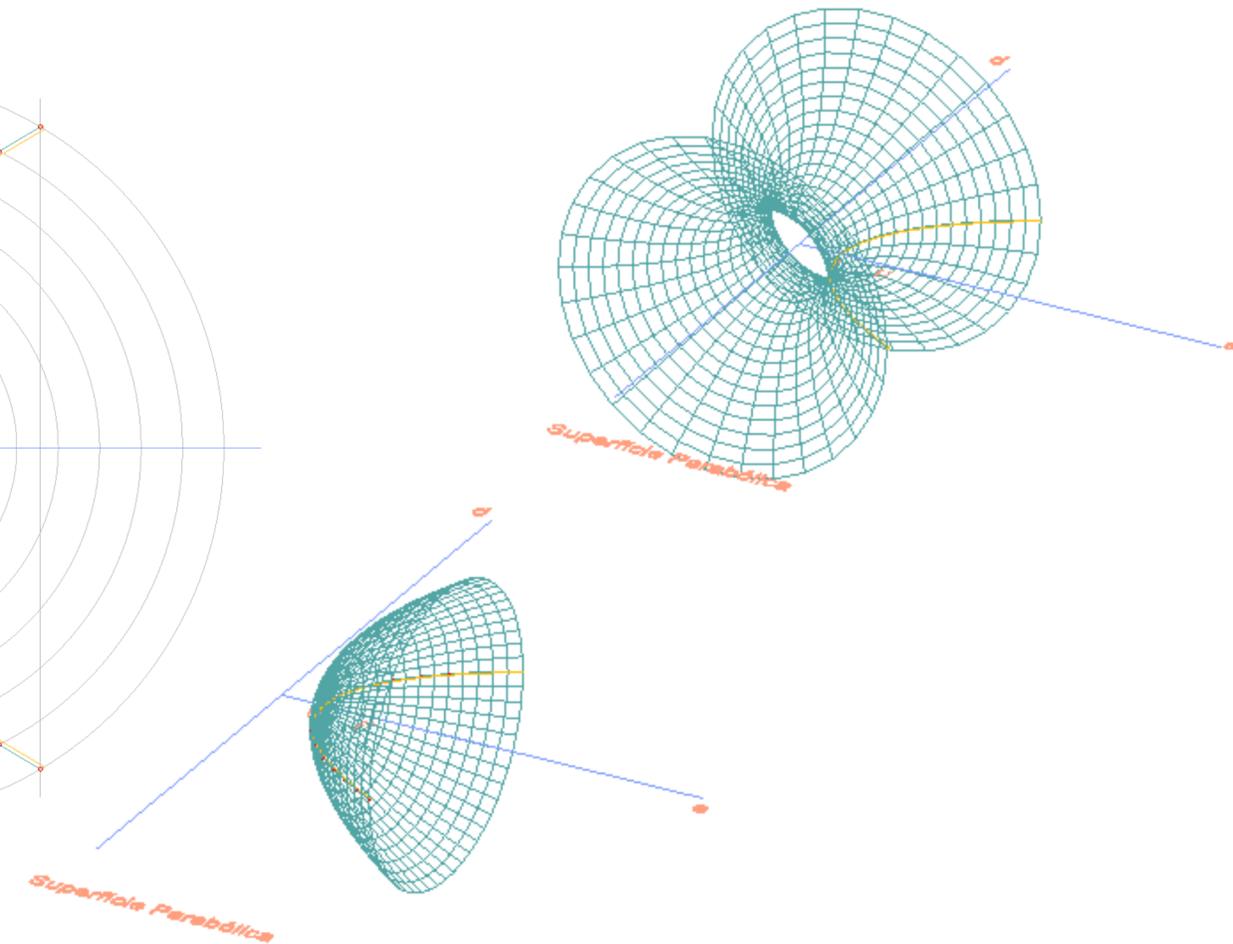
Comandos principais: OFFSET. PEDIT. JOIN. DTEXT. REVSURF. SURFTAB. SHADE.



Marcação do eixo – *e* – e diretriz – *d*. Estabelecimento do Foco – *F* – e do Vértice – *V*.  
Marcação dos pontos da parábola.



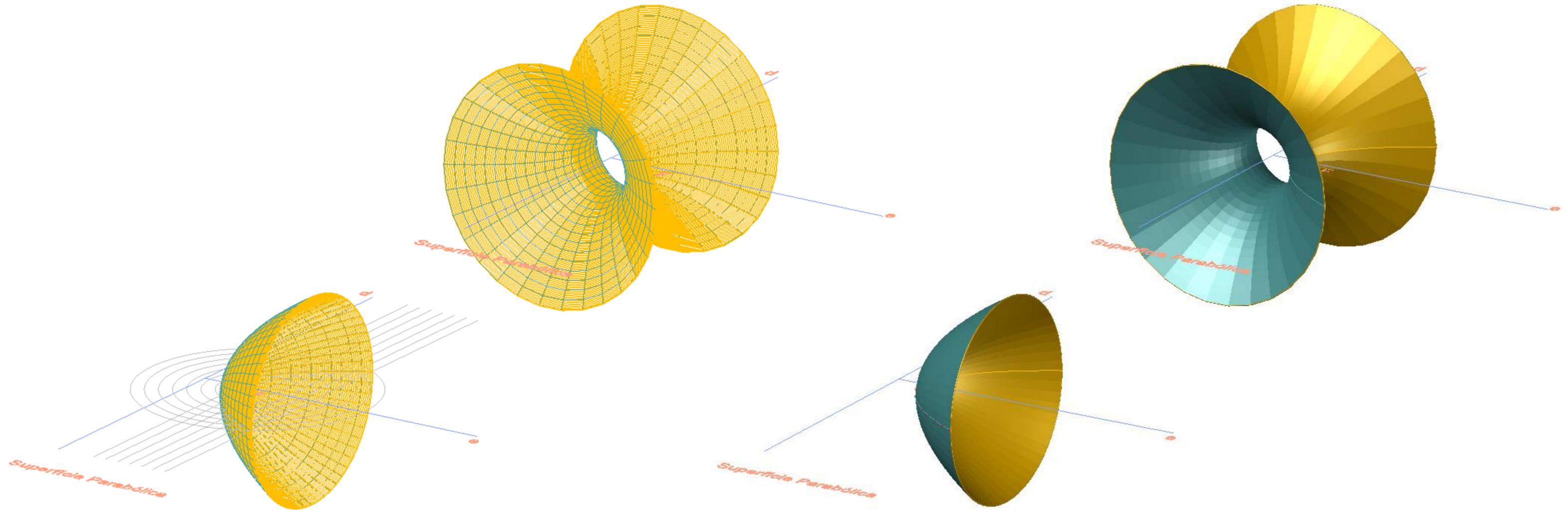
Marcação da parábola, unindo os pontos marcados.  
Criação de uma segunda parábola, paralela à primeira.  
Fechamento das parábolas, através de um semicírculo.



Revolução da parábola – Superfície Parabólica.

Em baixo, com um ângulo de revolução de 180°, com eixo em *e*, em cima, com um ângulo de 360°, com eixo em *d*.

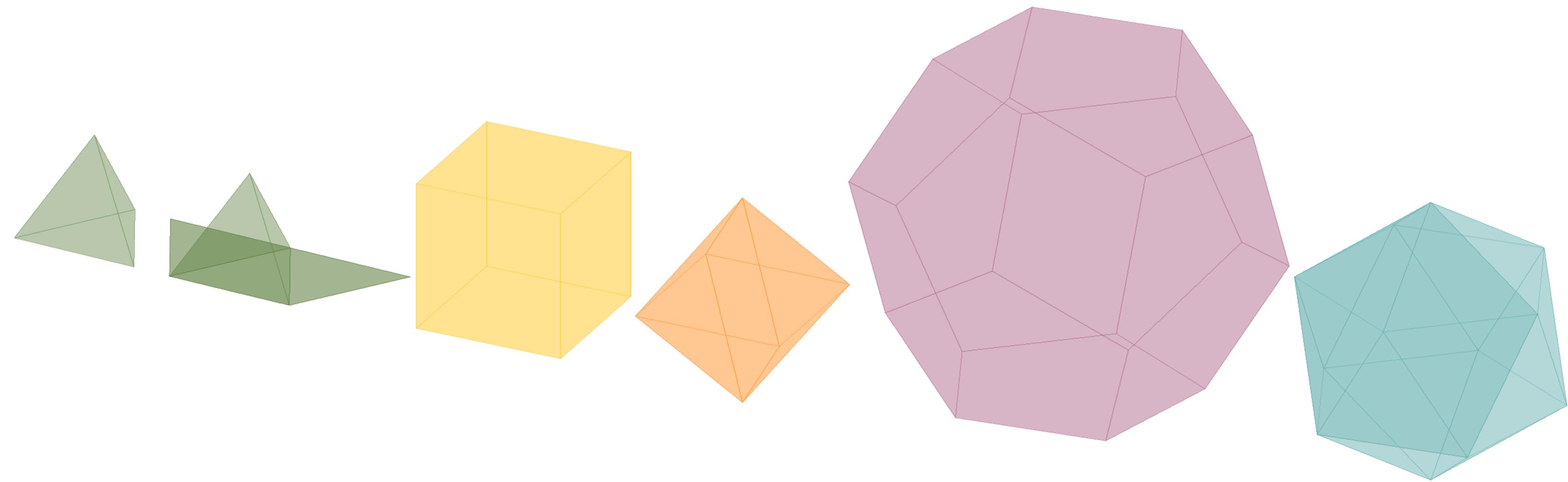
# Exerc. 1 - Superfície Parabólica



Revolução da parábola – *Superfície Parabólica*.  
Revolução da Segunda parábola, com fechamento.

Revolução da parábola – *Superfície Parabólica*.  
Comando **SHADE**.

# Exerc. 1 - Superfície Parabólica



# Exerc. 2 – Sólidos Platónicos

**Sólido Platônico** – poliedro cujas faces são polígonos regulares e congruentes entre si (os lados e os ângulos internos são iguais entre si).

O **Tetraedro** é um sólido platônico composto por quatro faces iguais – o resultado é uma pirâmide, cujas faces são triângulos equiláteros.

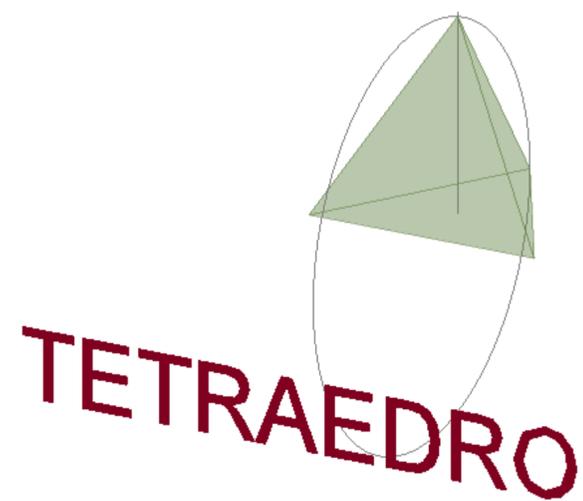
Com as quatro faces desenhadas sobre o *plano horizontal*, é necessário rebatê-las. Uma vez que as faces são triângulos equiláteros, a cada uma delas vai opor-se um vértice. Assim sendo, no *centro geométrico* da base vai erguer-se uma reta vertical – aonde se vai localizar o vértice comum a todas as faces laterais. Falta, portanto, determinar uma circunferência que vai servir de *charneira de rebatimento* – uma circunferência, com centro na aresta que a face lateral partilha com a base (A), e com raio ao vértice oposto. Utilizando o comando **3DROTATE** – utiliza um dado ponto e um dos eixos (x, y e z) como charneira – e utilizando como *charneira* o ponto A, ergue-se a circunferência. No ponto de interseção da circunferência criada com a reta vertical tem-se o vértice superior do tetraedro.

- Tendo as linhas auxiliares determinadas, usaram-se dois métodos diferentes:

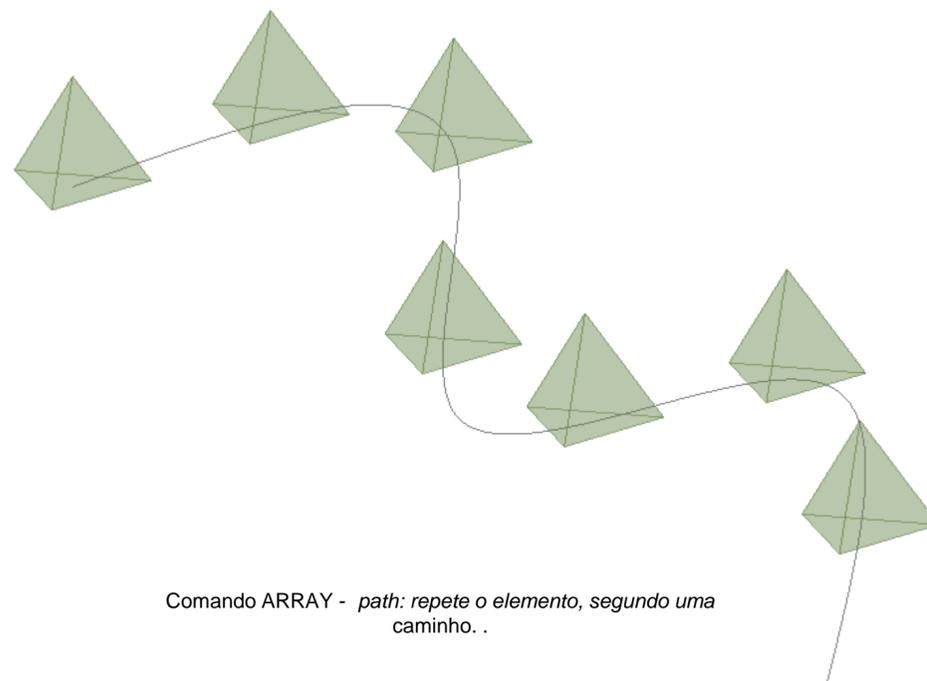
Numa *primeira abordagem*, utilizando o comando **3DROTATE**, as faces foram erguidas: estabeleceram-se como pontos de *charneiras* as *mediatrizes das arestas comuns às bases e faces laterais* – os vértices opostos a estas arestas são erguidos, formando o tetraedro.

- Numa segunda abordagem, foi usado o comando **ARRAY**. Erguendo-se a primeira face lateral, e escolhendo uma charneira, o comando permite *criar faces espelhadas, rodadas segundo a charneira*. Neste caso, pretendeu-se criar 3 faces laterais, usando como *charneira* o *centro geométrico da base*.

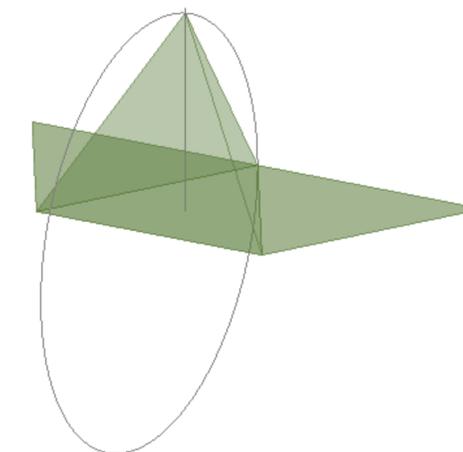
Comandos principais: DTEXT. 3DROTATE. MIRROR. ARRAY.



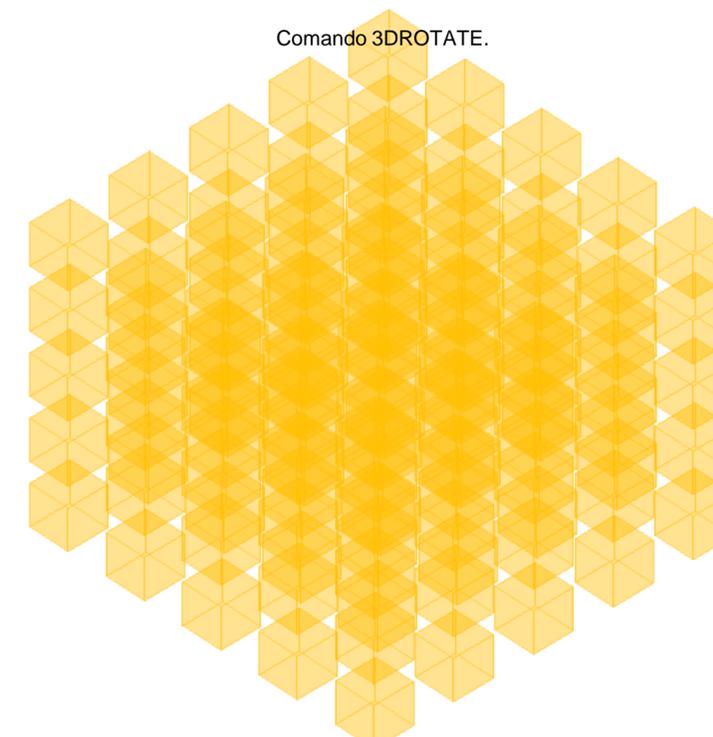
Comando ARRAY – polar: repete um dado elemento, segundo uma charneira.



Comando ARRAY - path: repete o elemento, segundo uma caminho. .



Comando 3DROTATE.



Comando ARRAY - rectangular: repete o elemento, segundo os eixos x, y e z.

# Exerc. 2.1.1 – Tetraedro

**Hexaedro** – sólido platónico composto por seis faces iguais – o resultado é um cubo (prima com faces quadrangulares).

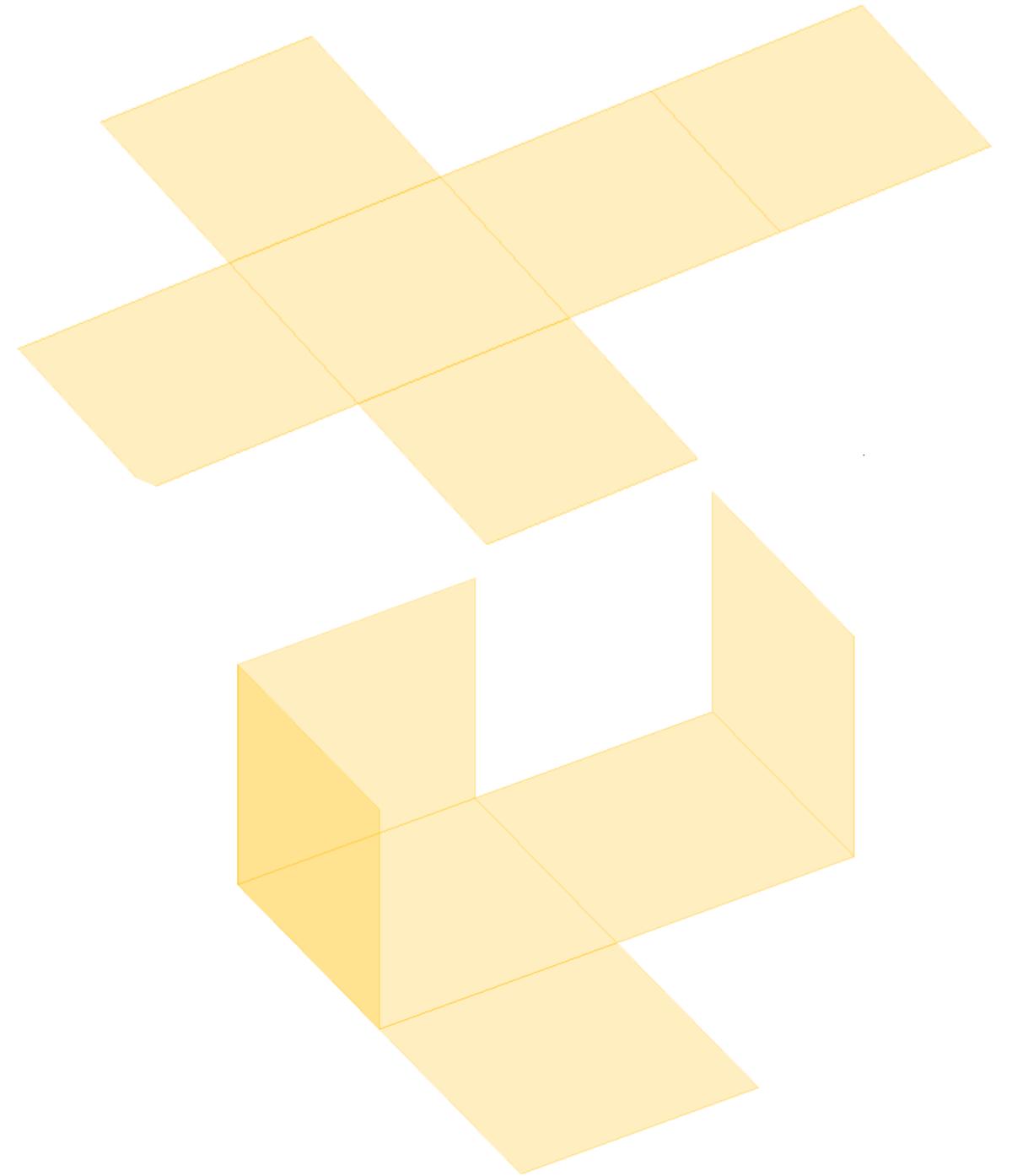
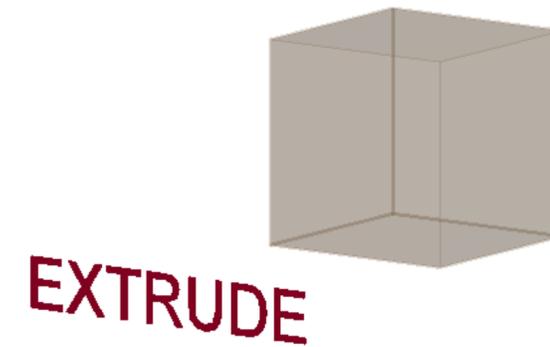
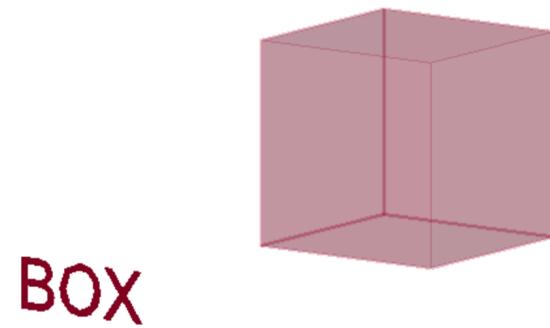
Para a modelação do hexaedro, e explorando dois novos comandos, recorreu-se a três abordagens diferentes.

- Com o comando **BOX**, estabelecendo um primeiro ponto, de partida, torna-se muito rápido modelar prismas recorrendo a este comando. Estabelecendo coordenadas relativas ao ponto inicial, definem-se os vértices restantes: determinando os restantes vértices, dimensionando as arestas, segundo os eixos *x*, *y* e *z* – neste caso, por ser um hexaedro, os valores relativos ao ponto inicial, segundo os eixos *x*, *y* e *z*, vão ser iguais.
- Com o comando **EXTRUDE**, a base quadrangular é erguida, *como que puxada*, formando um prisma. Definindo a altura do prisma (=aresta da base), obtém-se o hexaedro.
- Por fim, e a fim de pôr em prática o comando **3DROTATE**, rebateram-se as faces laterais e superior do hexaedro, que estavam rebatidas sobre o plano horizontal.

Aparentemente, os três métodos produzem resultados semelhantes: a grande diferença é que com os comandos **BOX** e **EXTRUDE**, o elemento criado é sólido, enquanto que ao utilizar o comando **3DROTATE**, o elemento resultante é oco, como uma caixa – uma vez que resulta de um processo de *levantamento de paredes*.

Seria, ainda, possível repetir o sólido através do comando **ARRAY**. O resultado seria oco, como acontece com a utilização do comando **3DROTATE**, com o rebatimento das faces.

Comandos principais: DTEXT. MIRROR. 3DROTATE. BOX. EXTRUDE.



Comando 3DROTATE – processo.

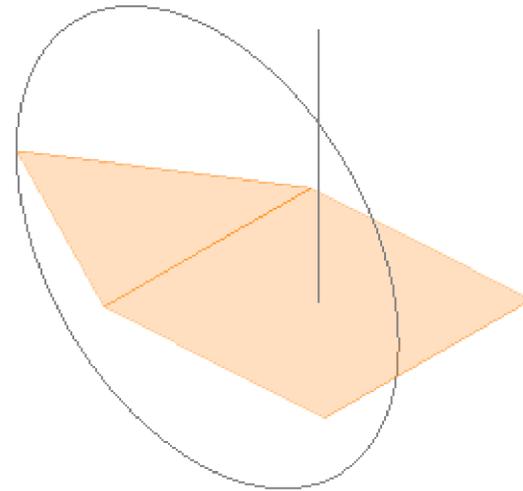
# Exerc. 2.1.2 – Hexaedro

**Octaedro** – sólido platônico composto por oito faces iguais – o resultado é uma agregação do que aparentam ser duas pirâmides espalhadas, cujas faces são triângulos equiláteros

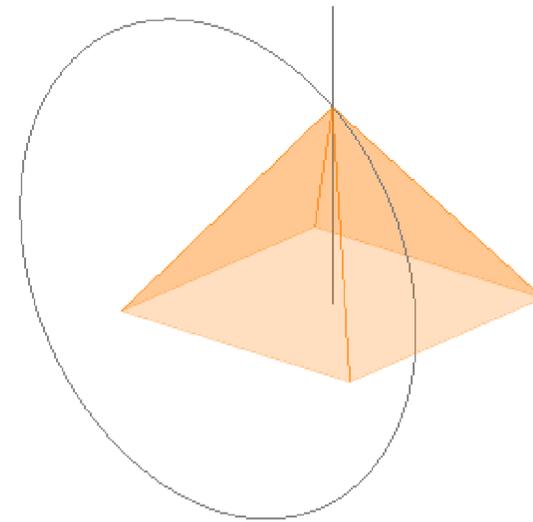
O Octaedro foi uma revisão de comandos anteriores. Estabelecendo uma base quadrangular, e desenhando uma face triangular rebatida sobre o plano horizontal, o processo utilizado para a construção do tetraedro repete-se. Achando o vértice da pirâmide, na interseção da circunferência com a reta vertical (que passa no *centro geométrico* da base), a primeira face é erguida. Para determinar as restantes faces da primeira pirâmide, é utilizado o comando **ARRAY**.

Apagando a base da pirâmide agora formada (uma vez que o octaedro não é dividido por nenhuma face ao centro), resta fazer um **MIRROR** das faces laterais: ou seja, espelhá-las, obtendo as restantes.

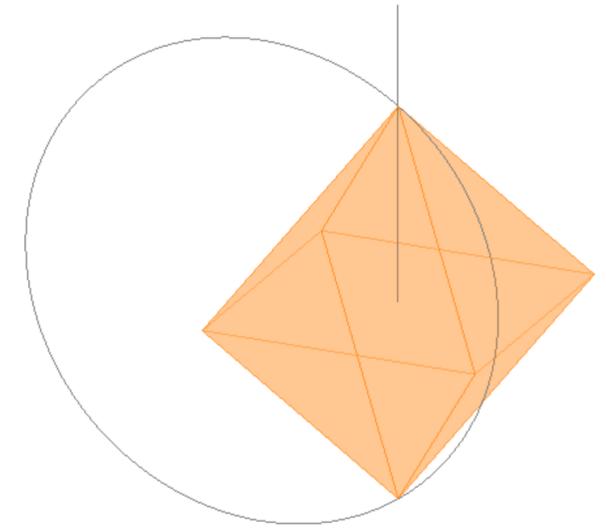
Comandos principais: DTEXT. 3DROTATE. BOX. EXTRUDE. ARRAY. MIRROR.



Comando 3DROTATE – processo.



Comando 3DROTATE – processo.



**OCTAEDRO**

# Exerc. 2.1.3 – Octaedro

O **Dodecaedro** é um sólido platônico composto por 12 faces iguais – as faces são pentágonos regulares.

Para a realização do dodecaedro, uma vez que este é um sólido de construção mais complexa que os anteriores, decidi dividir-se a figura em 2 partes iguais: em duas composições, *uma espécie de taça* que, espelhadas e rodadas em si, formariam o *dodecaedro*.

Sendo que todas as faces são pentágonos regulares, começou por se desenhar três pentágonos, rebatidos sobre o plano horizontal: uma das *bases* do sólido, e duas faces laterais. A maior dificuldade da construção tridimensional desta figura está em determinar o vértice superior, partilhado pelas faces laterais deste meio sólido.

Para este efeito, marcou-se uma reta no prolongamento da aresta que estas partilham com a base e uma reta perpendicular a esta, que passa no vértice **A** – como demonstrado nas figuras.

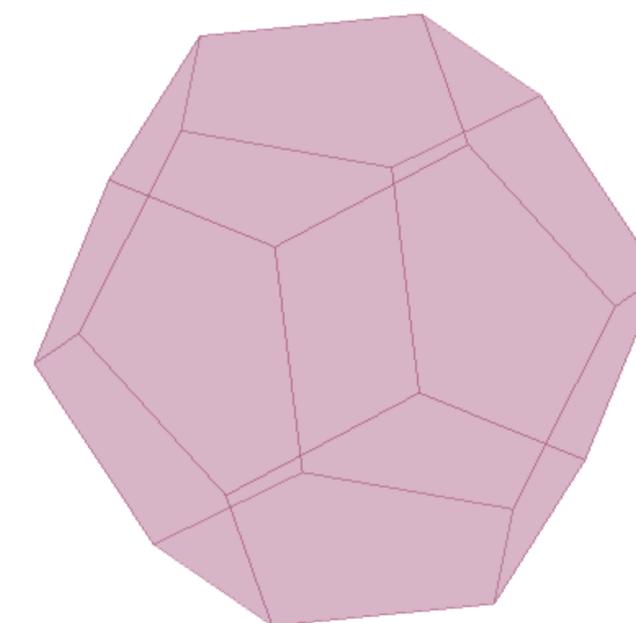
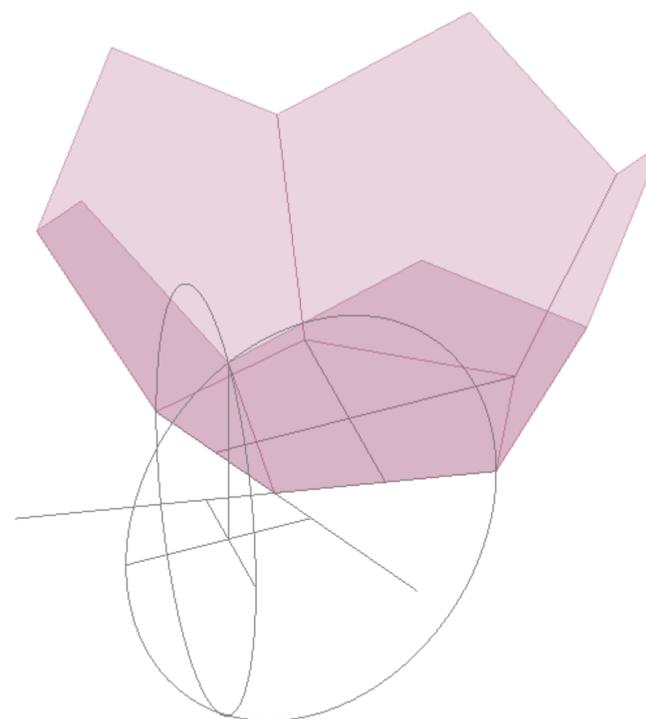
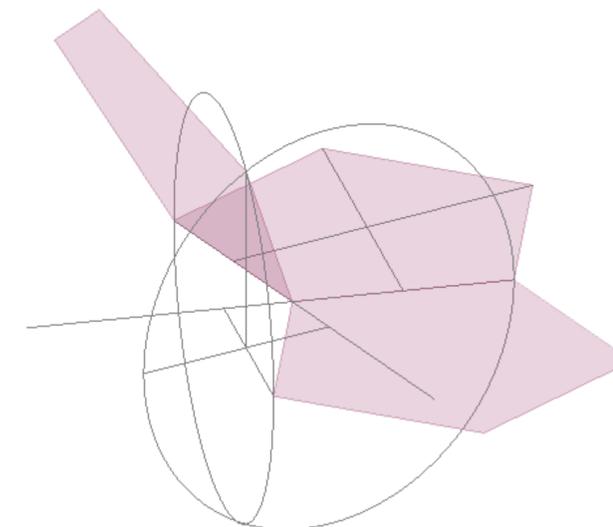
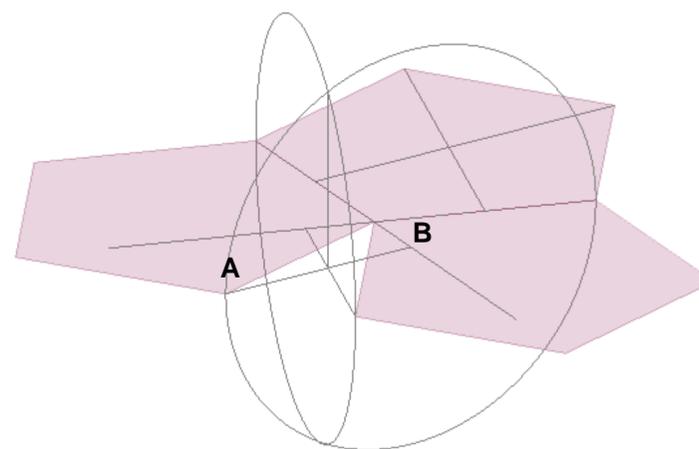
No ponto de interseção das duas retas, no ponto **B**, marca-se uma circunferência, com raio ao vértice **A**. Utilizando o comando **3DROTATE**, faz-se uma rotação da circunferência, para que esta permaneça ao alto – esta circunferência estabelece uma linha de rotação à face lateral – com *charneira* no ponto **B**.

Repete-se o processo para a outra face.

O ponto de interseção das retas perpendiculares ao prolongamento das arestas da base dos pentágonos laterais (que é a projeção no *plano x,y* do ponto de interseção das circunferências), corresponde ao vértice partilhados pelas faces laterais. Assim sendo, utilizando como *charneira* o ponto **B**, a face lateral é levantada – tendo determinada a inclinação de uma das faces laterais, utilizando o comando **ARRAY**, completa-se esta metade do octaedro.

Utilizando o comando **MIRROR**, espelha-se esta metade. Uma vez que os vértices da outra metade são desconstruídos com a primeira, de forma a estas metades encaixarem, é necessário alinhar os vértices. Isto pode conseguir-se com o recurso a dois comandos: com o comando **ALIGN**, os vértices alinham-se diretamente (escolhendo um ponto de partida e um ponto de chegada) ou o comando **3DROTATE**, rodando a figura sobre si mesma,  $36^\circ$  (uma vez que os vértices das bases se encontram a meio das arestas da base oposta e sendo que os ângulos internos de um pentágono são  $72^\circ$ , então as bases são rodadas  $36^\circ$  – metade do ângulo interno das figuras).

Comandos principais: 3DROTATE. MIRROR. ARRAY. ALIGN.



DODECAEDRO

Comando ARRAY - POLAR

# Exerc. 2.1.4 - Dodecaedro

**Icosaedro** – sólido platônico composto por 20 faces iguais – 20 triângulos equiláteros.

Para a construção do Icosaedro, decidi dividir-se o sólido em 3 partes: duas pirâmides pentagonais iguais, que formam o topo e base, e o corpo do icosaedro, o tronco do sólido.

Desta forma, e seguindo passos explorados para os outros sólidos, fizeram-se, primeiro, as pirâmides pentagonais, que constituem o topo e base do sólido. Uma vez que as faces são triângulos equiláteros, desenhou-se primeiro o pentágono e uma das faces. Sabendo que a figura pretendida era uma pirâmide regular, sabia-se que todas as faces laterais desta pirâmide coincidem num único vértice, num ponto de reta vertical traçada pelo centro da base pentagonal. Assim sendo, rebateu-se a primeira face lateral, segundo uma circunferência, até esta intersejar a reta vertical (utilizando o comando **3DROTATE**). Tendo uma face erguida, de seguida utilizou-se o comando **ARRAY**, completando a pirâmide.

Ao espelhar esta pirâmide (comando **MIRROR**), obteve-se a base do icosaedro. De seguida, resta apagar os pentágonos utilizados inicialmente na construção do topo e da base do sólido, uma vez que o icosaedro não é interjetado por nenhum plano.

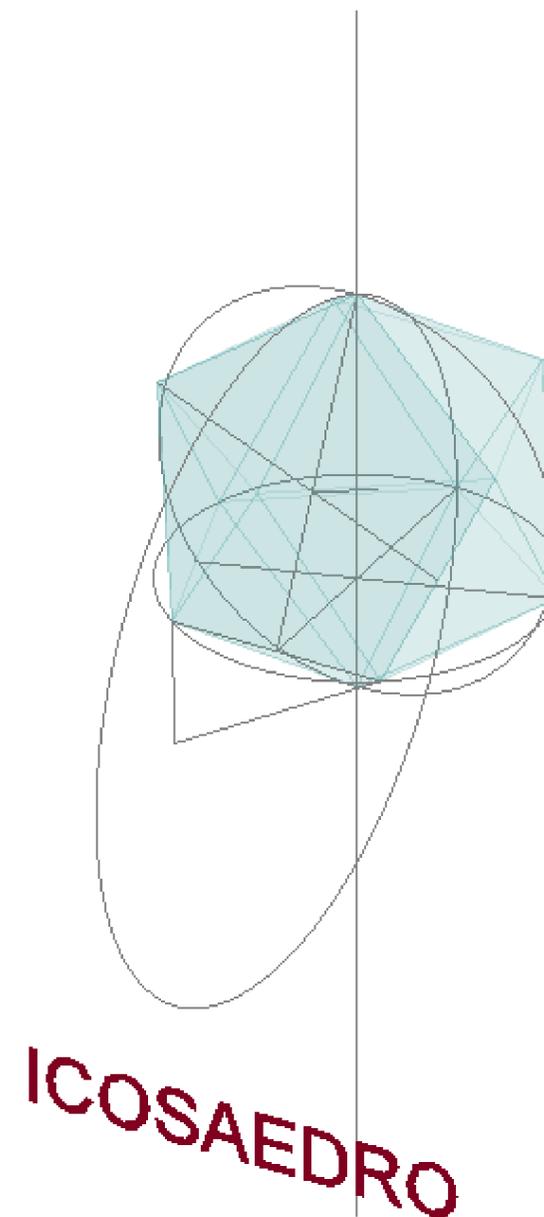
Para a construção do tronco do icosaedro, composto por 10 triângulos equiláteros, sabia-se à partida que a base e topo seriam rodados sobre si mesmos  $36^\circ$  - de forma a que, vistas do topo, estas pirâmides estariam desencontradas (como as faces do icosaedro são triângulos equiláteros, cada aresta do pentágono resultante das pirâmides vai corresponder a uma aresta dos triângulos equiláteros que compõem o tronco, e o vértice da pirâmide oposta, equidistante a cada um destes vértices será o terceiro vértice do triângulo – o tronco do icosaedro corresponde a um **ANTIPRISMA**).

Desta forma, para determinar a interseção deste vértice da face lateral com o vértice da base (que, no fundo, corresponde à altura do icosaedro) rebate-se uma circunferência com centro na base do triângulo que se pretende rebater. O ponto de interseção desta circunferência com uma reta vertical com origem no vértice da base do icosaedro (referido anteriormente) é o vértice pretendido.

Com o recurso ao comando **MOVE**, desloca-se a base do sólido, tendo determinado a altura do mesmo.

Resta só, com o recurso ao comando **ARRAY**, multiplicar a face lateral, e com o comando **MIRROR**, espelhar o resultado, de forma a obter as 10 faces laterais do icosaedro (é necessário rodar as faces resultantes sobre si mesmas,  $36^\circ$ , de forma a que estas encaixem corretamente no sólido).

Comandos principais: 3DROTATE. MIRROR. ARRAY. MOVE.

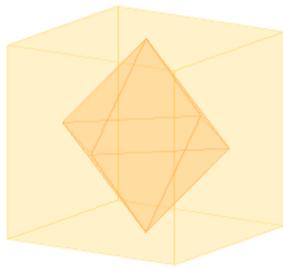
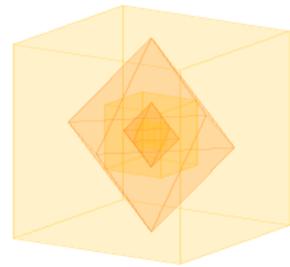


# Exerc. 2.1.5 - Icosaedro

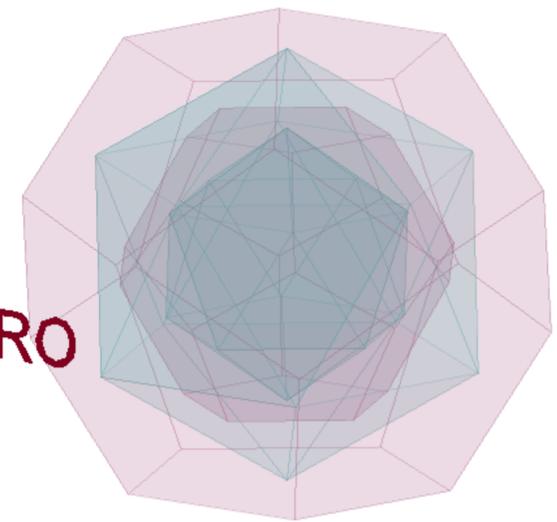
TETRAEDRO  
AUTOdual



HEXAEDRO OCTAEDRO  
RELAÇÃO DE DUALIDADE - duais



DODECAEDRO ICOSAEDRO  
RELAÇÃO DE DUALIDADE - duais



# Exerc. 2.2 – Relações de Dualidade

Sólido Platônico	Faces	Vértices
<i>Tetraedro</i>	4	4
<i>Hexaedro</i>	6	8
<i>Octaedro</i>	8	6
<i>Dodecaedro</i>	12	20
<i>Icosaedro</i>	20	12

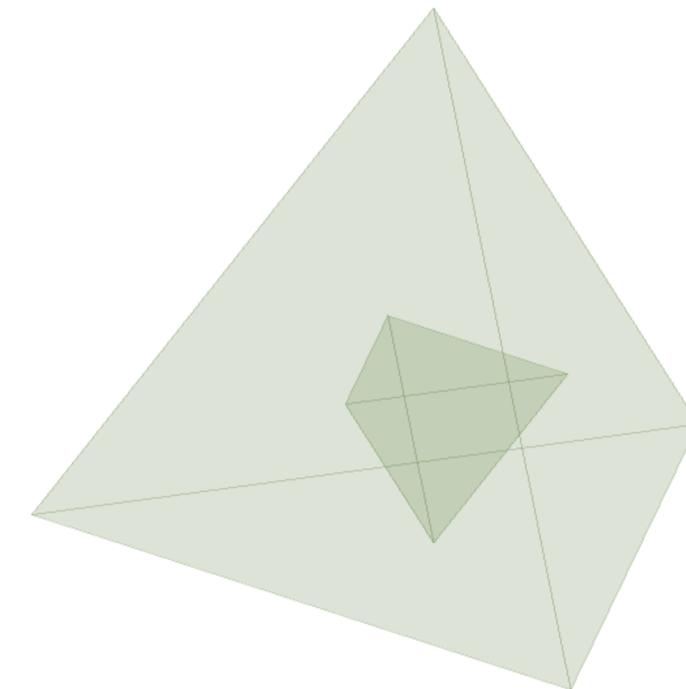
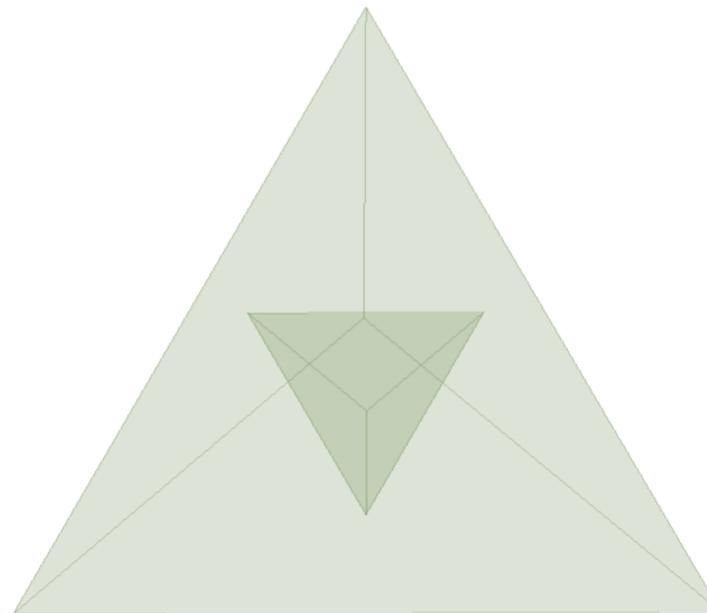
Com o recurso aos sólidos platônicos desenhados anteriormente, exploraram-se as relações duais formadas entre si.

**Relações de Dualidade** – ocorrem entre sólidos platônicos. Analisando o número de vértices e faces de cada sólido, é possível emparelhá-los (inserindo-os uns dentro dos outros), de forma a que a cada face corresponda um vértice (ou seja, os sólidos que tenham o mesmo número de vértices e faces).

O tetraedro é um **SÓLIDO AUTODUAL**, ou seja, tem o mesmo número de vértices e faces (4), sendo possível emparelhá-lo consigo mesmo (é o único sólido platônico **AUTODUAL**).

Tendo dois tetraedros desenhados, com recurso ao comando **ALIGN**, os vértices de um dos tetraedros são alinhados, ordenadamente, com o centro geométrico do segundo (resultando num tetraedro dentro de outro).

Comandos principais: COPY. ALIGN.



Comando ALIGN.

# Exerc. 2.2.1 – Tetraedro

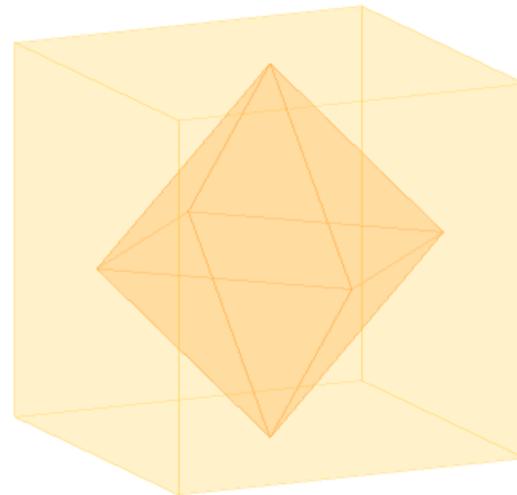
Sólido Platônico	Faces	Vértices
<i>Tetraedro</i>	4	4
<i>Hexaedro</i>	6	8
<i>Octaedro</i>	8	6
<i>Dodecaedro</i>	12	20
<i>Icosaedro</i>	20	12

Como é possível verificar na tabela acima, o hexaedro e o octaedro, partilham o mesmo número de faces e vértices (cruzados), ou seja, são **SÓLIDOS DUAIS**.

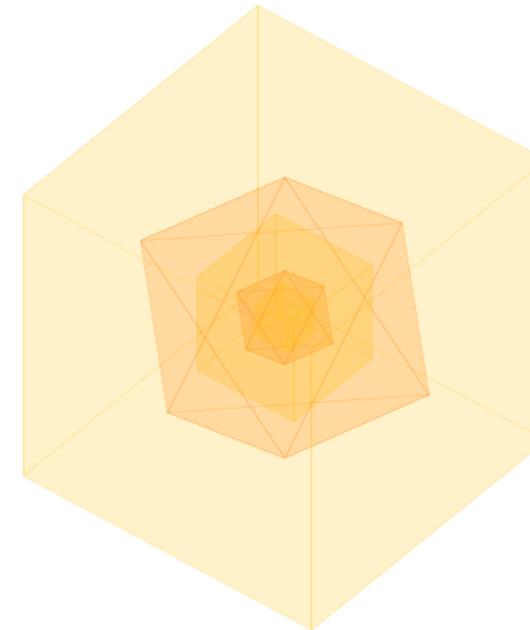
Assim sendo, repetiu-se o processo de alinhamento dos vértices com o centro geométrico das faces, com recurso ao comando **ALIGN**, o número de vezes pretendido.

A ordem pela qual os sólidos são alinhados é, como é possível entender, indiferente.

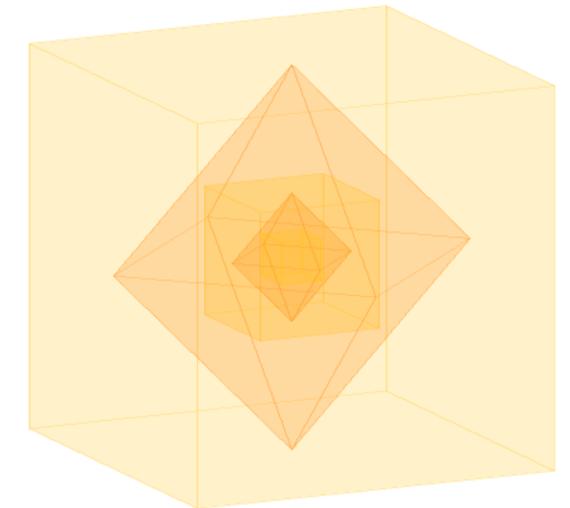
Comandos principais: COPY. ALIGN.



Comando ALIGN.



Cópia do resultante e comando ALIGN.



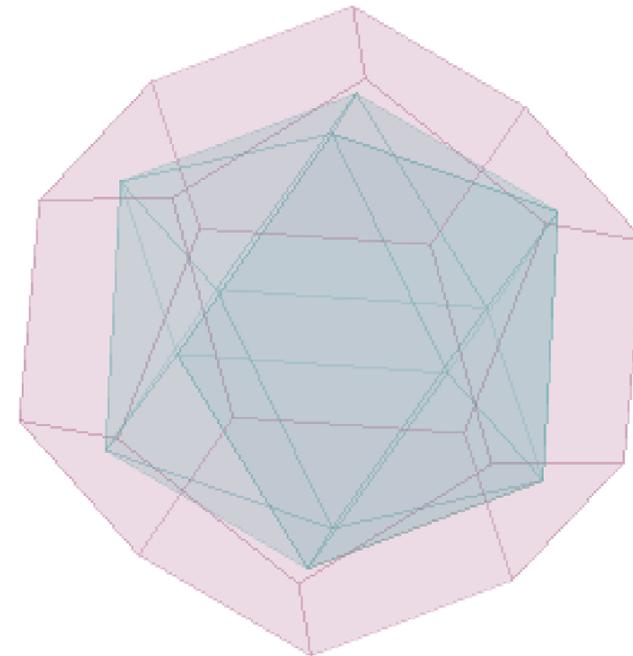
# Exerc. 2.2.2 – Hexaedro | Octaedro

Sólido Platónico	Faces	Vértices
<i>Tetraedro</i>	4	4
<i>Hexaedro</i>	6	8
<i>Octaedro</i>	8	6
<i>Dodecaedro</i>	12	20
<i>Icosaedro</i>	20	12

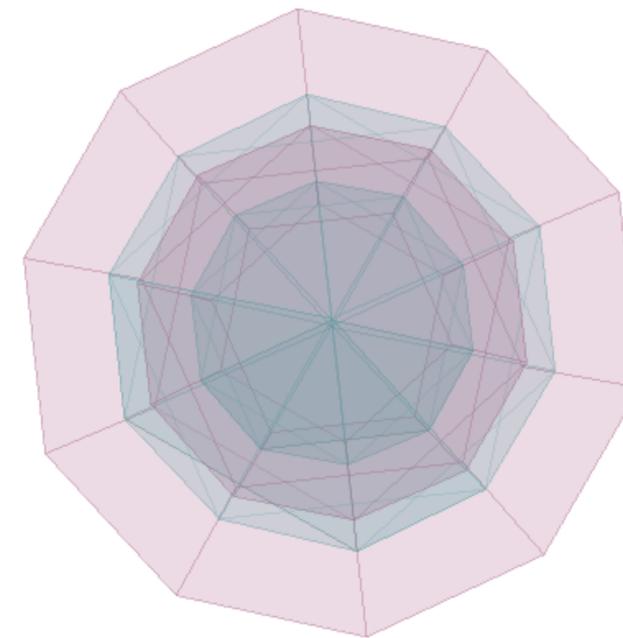
Da mesma forma, o dodecaedro e o icosaedro partilham, entre si, o mesmo número de vértices e faces (cruzados), pelo que, são **SÓLIDOS DUAIS**.

Repetiu-se, mais uma vez, o processo de alinhamento, com recurso ao comando **ALIGN**.

Comandos principais: COPY. ALIGN.



Comando ALIGN.



Cópia do resultante e comando ALIGN.

# Exerc. 2.2.3 – Dodecaedro | Icosaedro

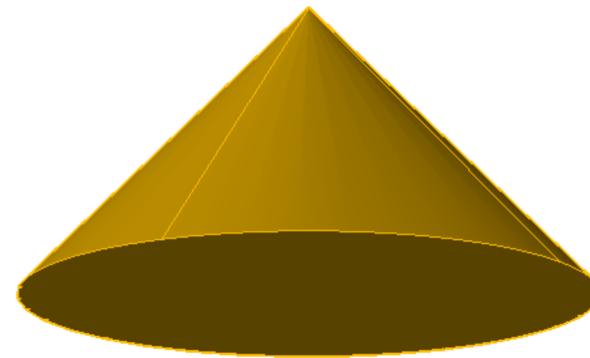
O exercício consiste no desenho de dois cones (superfícies cónicas), empilhados, de forma a que, pela introdução de planos secantes, resultem as devidas secções.

Para o desenho dos cones foi utilizado o comando **CONE**, especificando o raio e altura pretendidos. Uma vez que o objetivo era que este cone fosse algo como uma superfície, não sendo completamente preenchido, desenhou-se um segundo cone (uma cópia do primeiro), desenhado dentro do primeiro (com um ligeiro desfasamento).

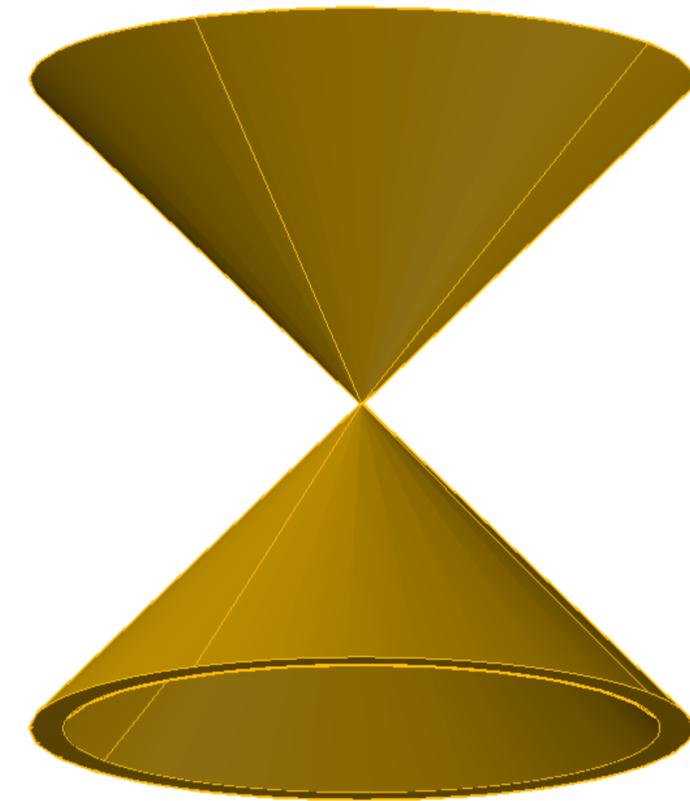
Com recurso ao comando **SUBTRACT**, subtrai-se o segundo cone ao primeiro, resultando uma superfície cónica.

Com o comando **MIRROR**, desenha-se um espelho da superfície resultante (o objeto final são duas superfícies cónicas, empilhadas).

*Comandos principais:* CONE. COPY. SUBTRACT. MIRROR.



Comando CONE.



Superfícies resultantes.

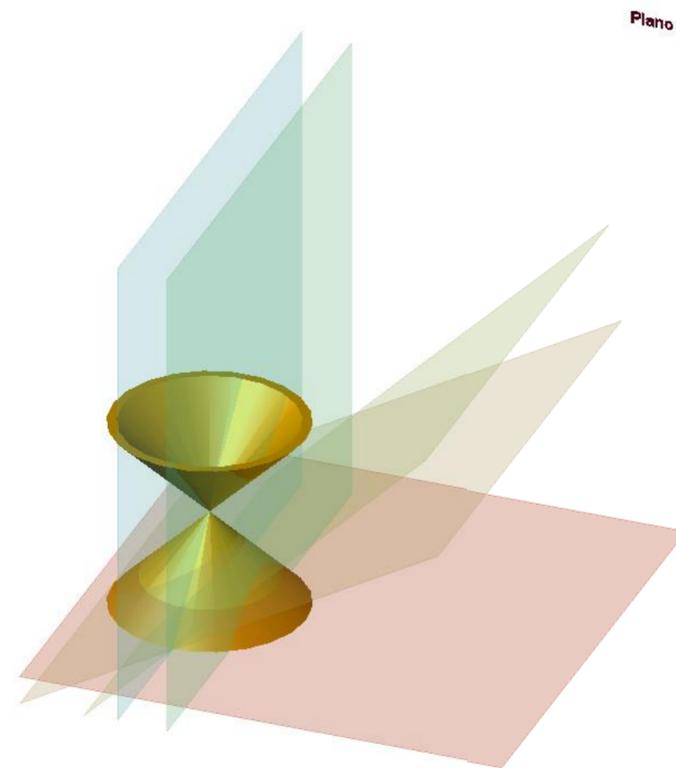
# Exerc. 3 – Cone

Tendo as superfícies cónicas desenhadas, pretende-se desenhar um conjunto de planos secantes, de diferentes inclinações, de forma a obter diferentes secções.

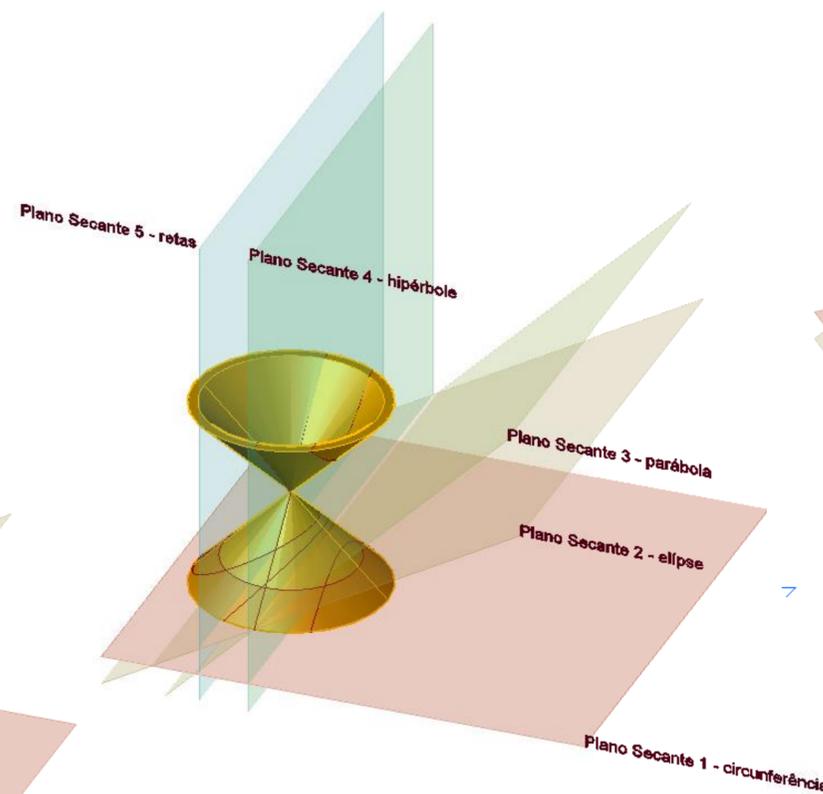
Para isto, desenha-se um primeiro plano, com recurso ao comando **PLANESURF**, e copia-se este mesmo plano 4 vezes. Cada uma destas cópias será direccionada para uma layer diferente, com diferentes cores (de forma a serem facilmente distinguíveis), e a cada uma delas é atribuída uma dada rotação e posicionamento (definidos).

Com recurso ao comando **SECTION**, as superfícies cónicas são seccionadas, originando as secções pretendidas.

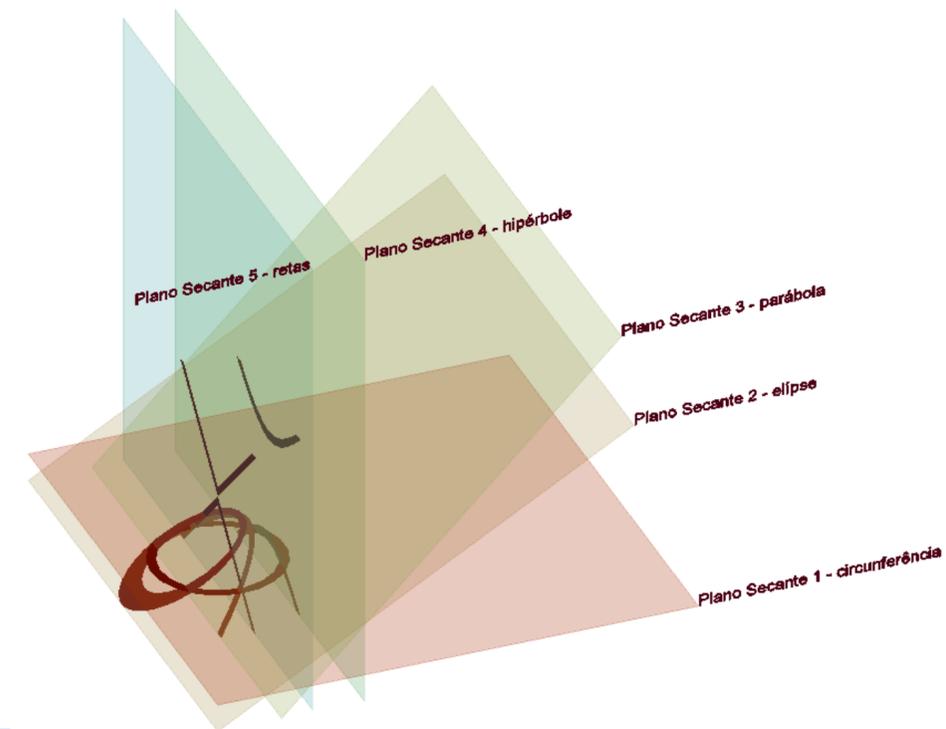
Comandos principais: PLANESURF. COPY. 3DROTATE. MOVE. SECTION.



Comando PLANESURF, 3DROTATE e MOVE.



Comando DTEXT e SECTION.

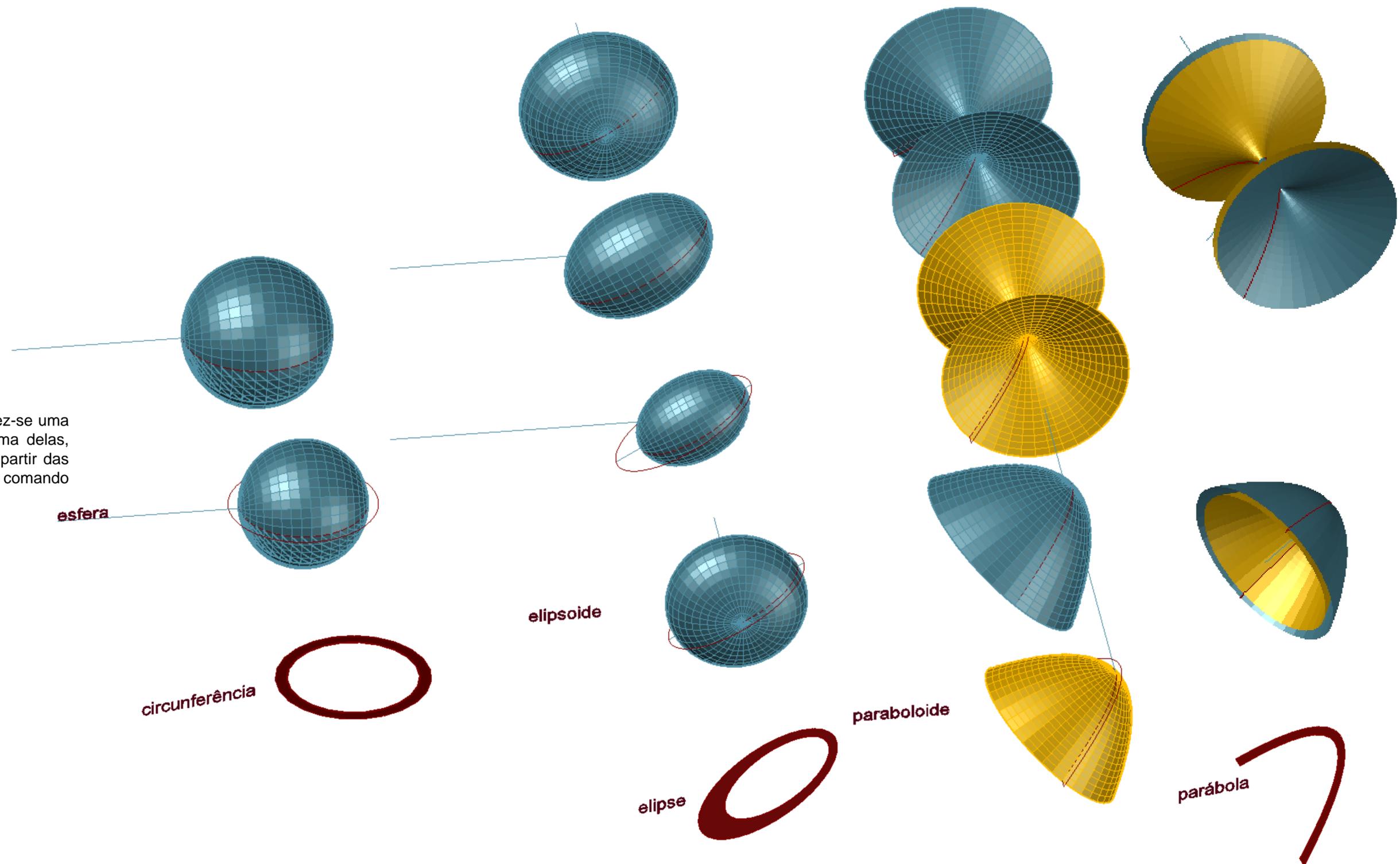


Secções Resultantes.

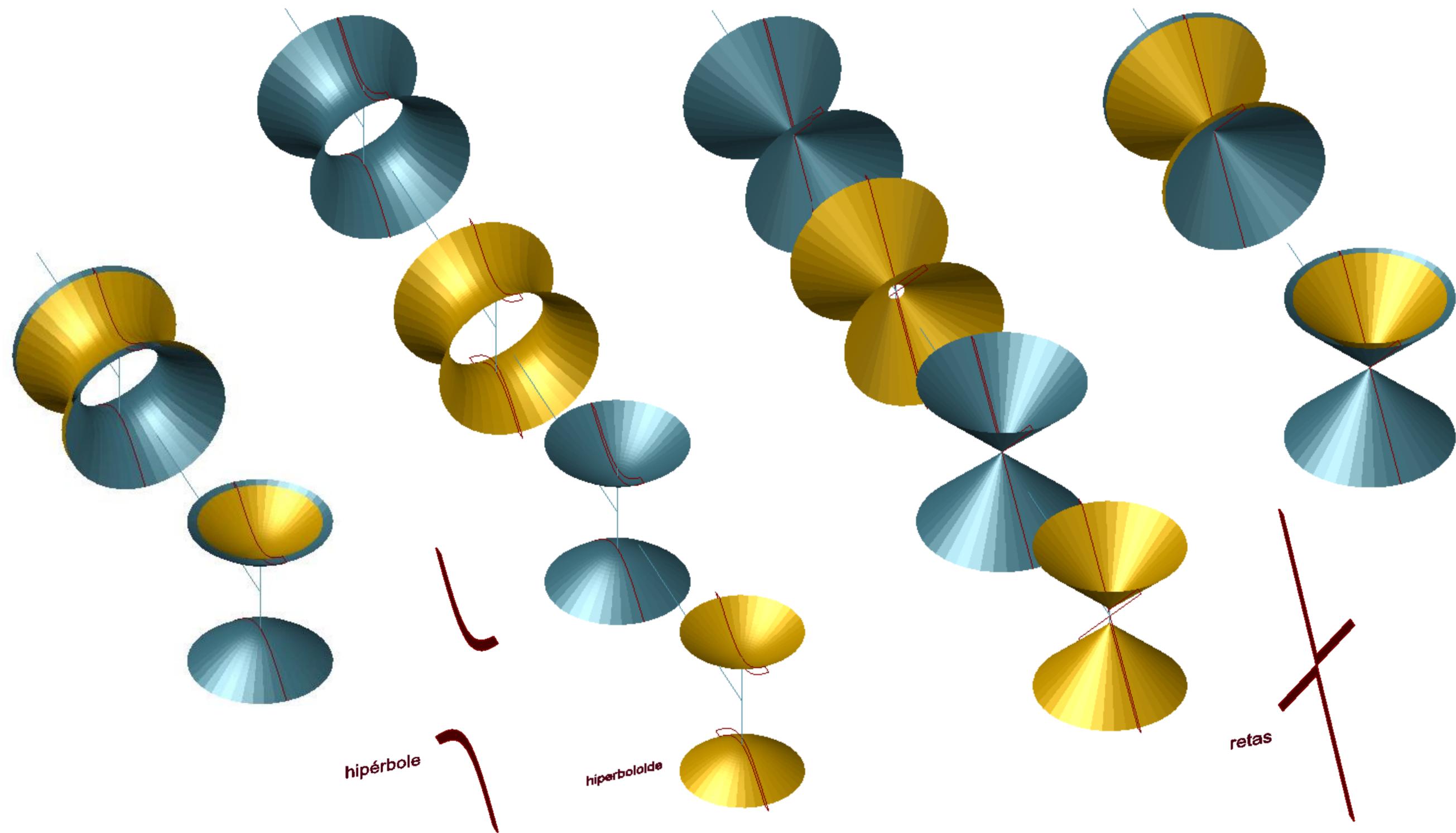
# Exerc. 3.1 – Inters. Planos Secantes

Tendo as secções resultantes determinadas, fez-se uma cópia de cada uma delas e desenhou-se, em cada uma delas, uma reta paralela aos eixo y e eixo x- o objetivo é, a partir das secções, originar diferentes sólidos (com recurso ao comando **REVSURF**).

Comandos principais: REVSURF.

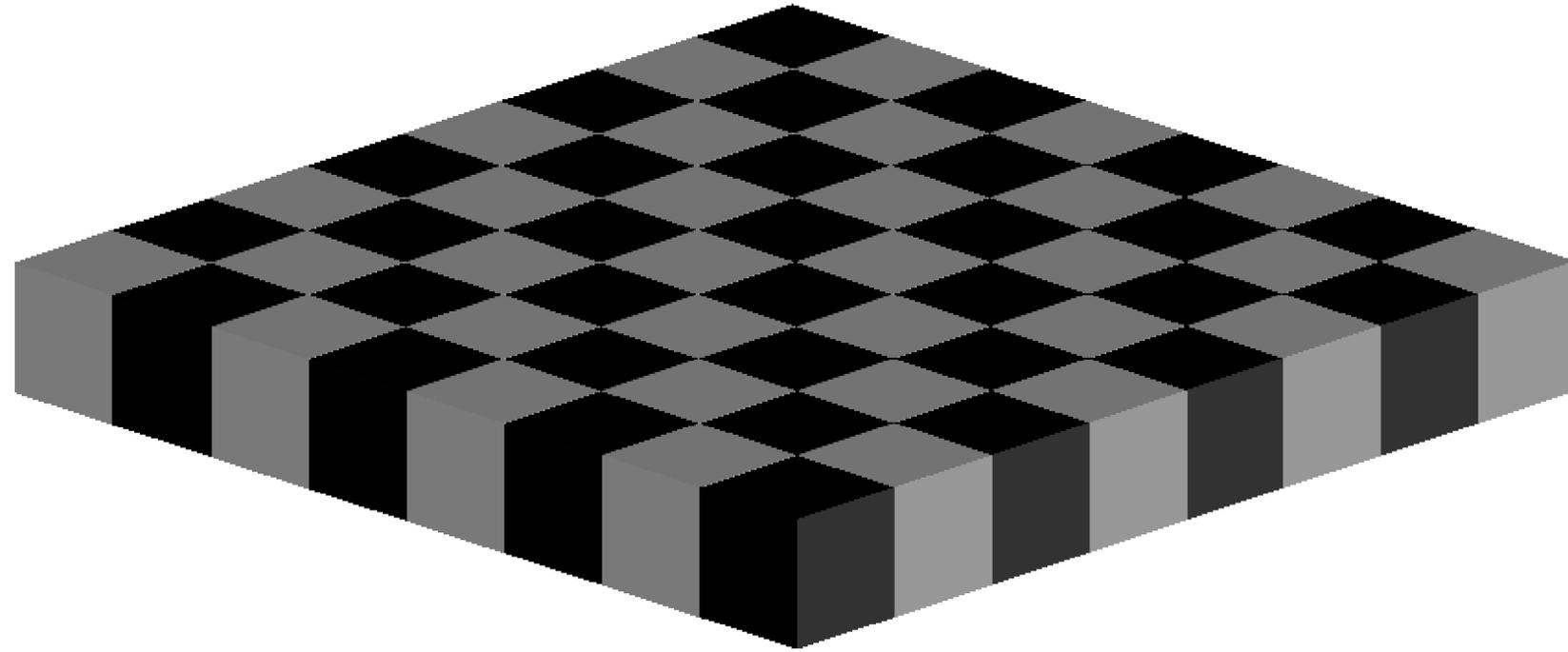


# Exerc. 3.2 – Superfícies Resultantes



# Exerc. 3.2 – Superfícies Resultantes

```
(Defun c:Xad ()  
(Command "box" "0,0,0" "10,10,10")  
(command "copy" "last" "" "0,0" "10,10")  
(command "mirror" "all" "" "10,0" "10,10" "n")  
(command "chprop" "previous" "" "c" "9" "")  
(command "array" "all" "" "r" "4" "4" "20" "20")  
)
```



Exerc. 4 - Xadrez

O exercício 4 consiste na **aprendizagem da programação de comandos**, passíveis de serem utilizados em **AUTOCAD**. Ou seja, a partir de um programa de programação, escrevem-se códigos, em sequência (e respeitando todas as indicações esperadas pelo programa final), de forma a criar um novo comando. Quando aplicado no **AUTOCAD**, o programa reconhece imediatamente o novo comando ( eliminando, desta forma, a necessidade de passos preliminares). *Esta é a vantagem da programação: criar novos comandos, que facilitam a tarefa do desenho técnico.*

Desta forma, utilizou-se o programa **NOTEPAD++**, como meio de criação de um novo comando: um comando denominado **Xad**, que, quando acionado no **AUTOCAD**, dará origem a um tabuleiro de xadrez.

Como sempre, para a programação, *todas as direções para este comando devem estar entre parênteses*, de forma a serem considerados válidos. A primeira frase *Defund c:*, garante a denominação pretendida para o novo comando: *o parênteses que antecede a primeira frase vai albergar todos os comandos aplicados, precisando de ser fechado, após todos os comandos terem sido programados, de forma a ser assumido como válido.*

Uma vez que o objetivo deste exercício era criar um tabuleiro de xadrez, o primeiro comando consiste na criação do comando “**box**”, seguido pelas coordenadas do ponto inicial (0,0,0), e dos restantes pontos necessários à formação do cubo – neste caso, os pontos que precisam de ser definidos encontram-se à distância de 10 metros, segundo o eixo x, y e z, respetivamente – *é de lembrar que para adicionar comandos, estes devem ser inseridos em frases individuais, entre parênteses*, e o comando pretendido, bem como todos os passos requeridos pelo **AUTOCAD**, devem ser *indicados entre aspas*.

Utilizando o comando **COPY**, copia-se o primeiro cubo (o segundo cubo criado também será um cubo branco – corresponde a uma casa de xadrez branca, pelo que para a cópia do primeiro cubo é necessário ter em atenção que os cubos da mesma cor aparecerão intercalados) – para acionar o **ENTER**, utilizam-se duas aspas seguidas, “”.

Para obter os cubos pretos, utiliza-se o comando **MIRROR**, espelhando os cubos já obtidos, e define-se uma reta charneira.

Com o comando **CHPROP**, é possível mudar as características dos elementos (*layer, cor, tipo de linha,...*). Respondendo às variantes, seleciona-se a propriedade da cor, **c**, e a cor que se pretende.

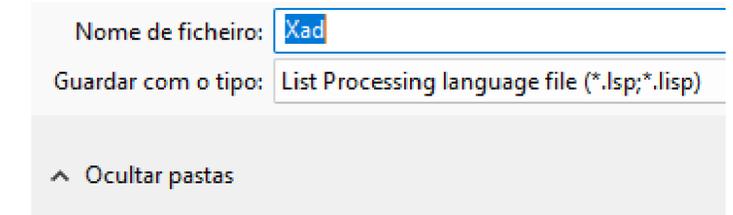
Resta multiplicar os 4 cubos desenhados, com recurso ao comando **ARRAY** (*rectangular*), definindo as variantes pretendidas. Fechando o comando, com um *último parênteses*, resta guardar como *ficheiro .lisp*, e importar o ficheiro para o **AUTOCAD**, com o recurso ao comando **APPLOAD** – com o comando importado, o **AUTOCAD** irá assumir o novo comando, pela sua denominação.

```

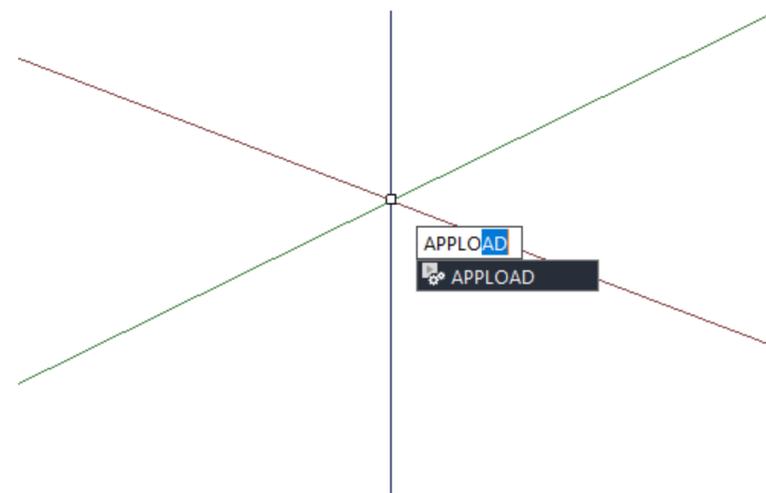
1 (Defun c:Xad ()
2 (Command "box" "0,0,0" "10,10,10")
3 (command "copy" "last" "" "0,0" "10,10")
4 (command "mirror" "all" "" "10,0" "10,10" "n")
5 (command "chprop" "previous" "" "c" "9" "")
6 (command "array" "all" "" "r" "4" "4" "20" "20")
7 )
8
9
10
11

```

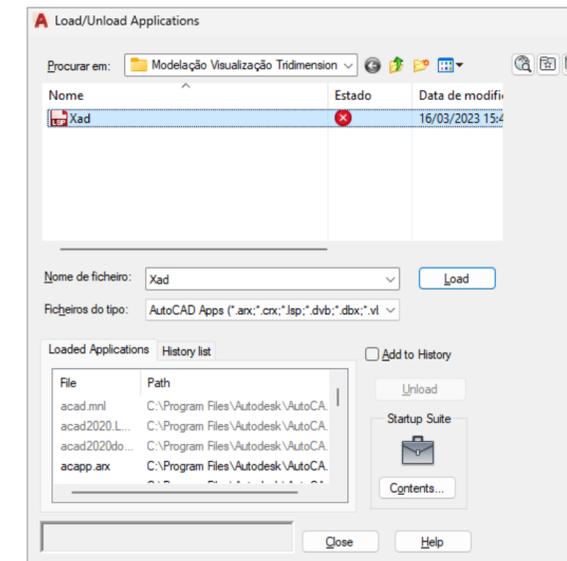
NOTEPAD++.



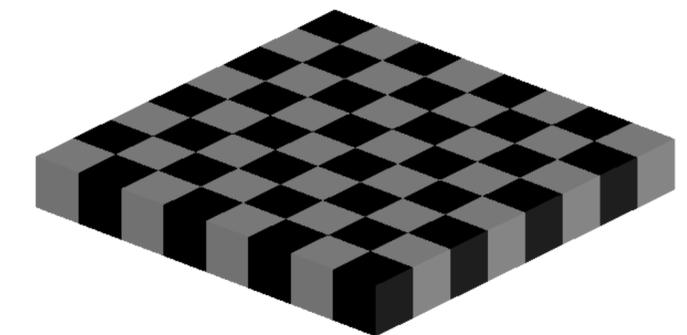
Guardar o ficheiro como ficheiro **LISP**.



Comando **APPLOAD**.

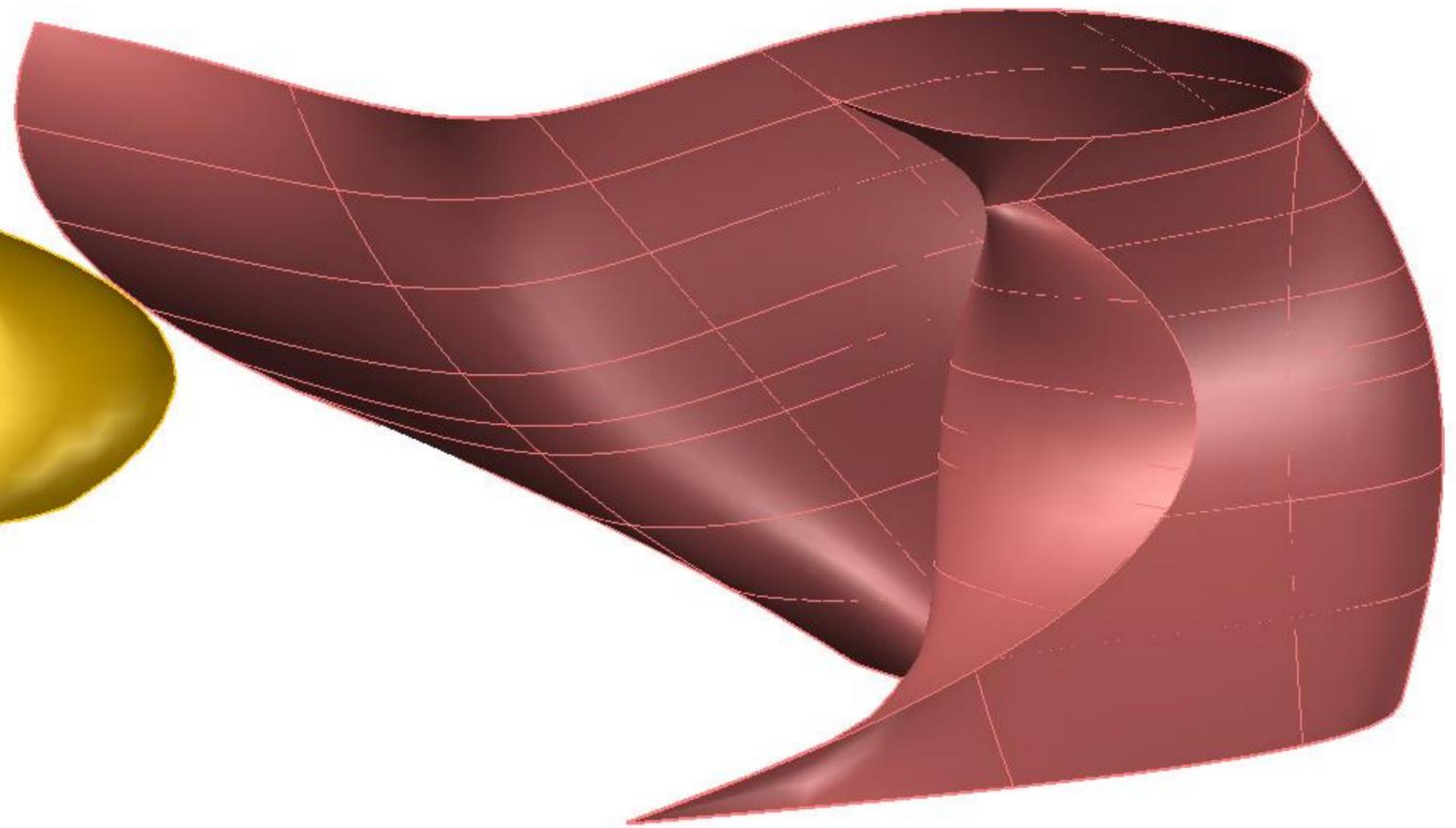
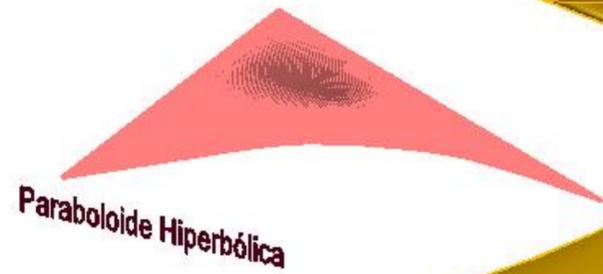


Comando **APPLOAD**.



Novo comando **XAD**.

# Exerc. 4 - Xadrez



Exerc. 5 – Hiperboloide e Comandos EDGESURF e *LOFT*

Partindo das secções cónicas planas, do terceiro exercício, pretendeu criar-se hiperboloides, determinando e manipulando as suas *geratrizes* – retas que determinam o plano da superfície.

Para comparação de resultados, cria-se uma primeira superfície hiperboloide, utilizando o comando **REVSURF**, e definindo como eixo a mediatriz das secções. Com o comando **SURFTAB**, manipula-se o número de planos curvos que compõe a superfície.

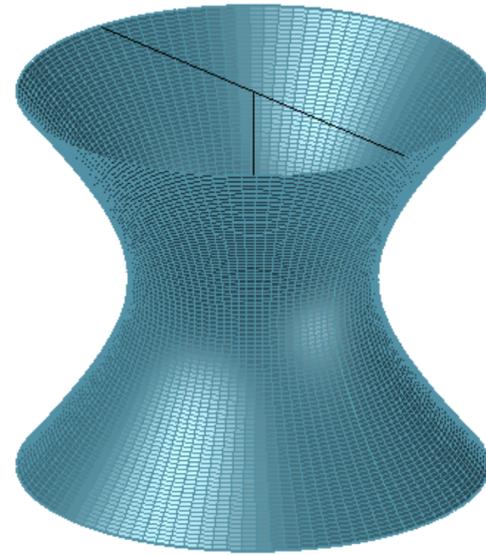
Assim sendo, determinam-se as circunferências delimitadoras da hiperboloide: pode considerar-se que esta superfície é *uma construção de sucessivas circunferências empilhadas*, estreitando na direção ao meio da figura (e, por sua vez, alargando em direção às extremidades). Desta forma, determinam-se as *circunferências maiores* da hiperboloide, nas extremidades, e a *circunferência menor*, na metade da figura. Uma vez que a *geratriz é uma reta frontal*, que interseca as circunferências criadas num ponto, marca-se uma paralela à circunferência menor sobre o *plano horizontal x,y* (plano que contém a circunferência da base). Marcando uma *tangente à circunferência menor*, determinam-se os dois pontos caracterizadores da geratriz (é a projeção horizontal da geratriz – uma *reta frontal*). Uma vez que o pretendido é criar *losangos* (delimitadores de planos que, em conjunto, poderão delimitar a superfície da hiperboloide).

Ao multiplicar as geratrizes determinadas, com o recurso ao comando **ARRAY** (*polar*), determinam-se geratrizes com a mesma inclinação e sentido das primeiras – *quanto maior for o número de retas, maior será a precisão da superfície* (é crucial, para ser passível de comparação, que o *número de itens seja igual ao número de planos estabelecido no SURFTAB*, para a criação da primeira superfície hiperboloide).

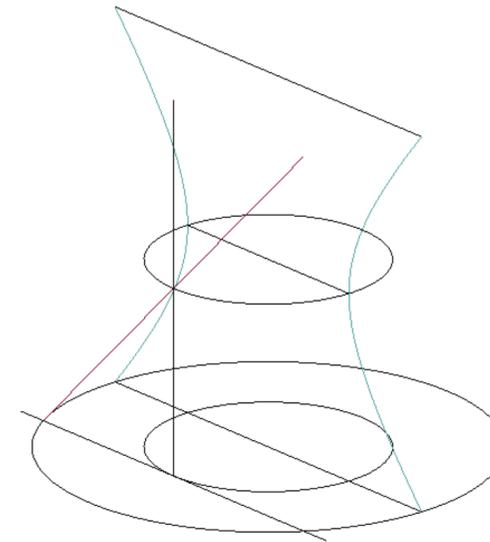
Aplicando o comando **SHADE**, os losangos são preenchidos por planos, criando a superfícies da hiperboloide.

Utilizando o comando **THICKEN**, atribui-se uma dada espessura à superfície hiperboloide, passando esta a ser um sólido.

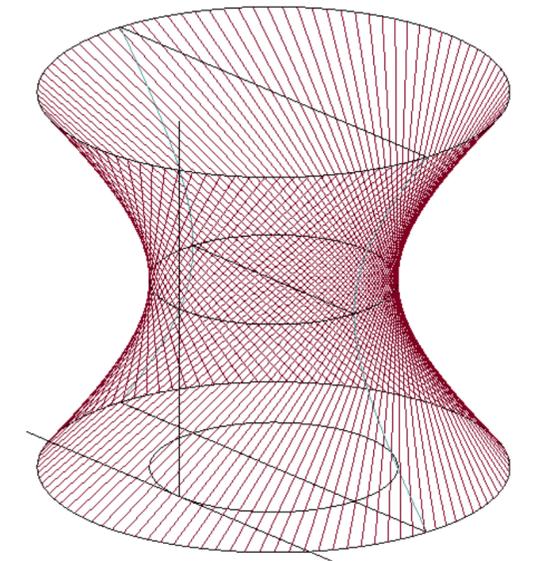
Comandos principais: REVSURF. SURFTAB. MIRROR. ARRAY.  
SHADE. THICKEN.



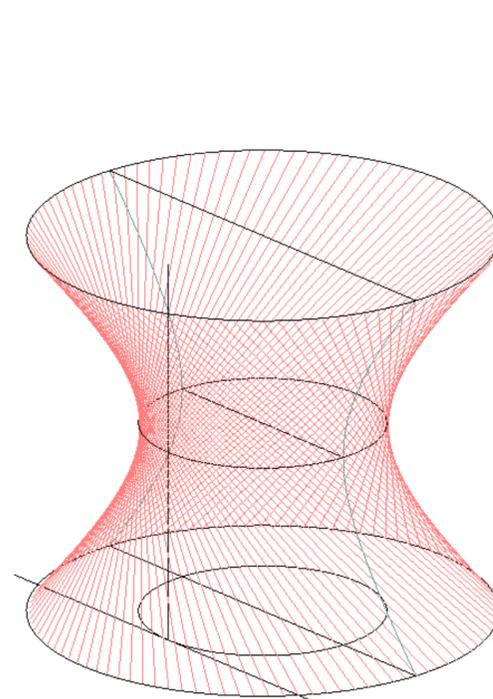
1ª Superfície Hiperbólica – comando REVSURF.



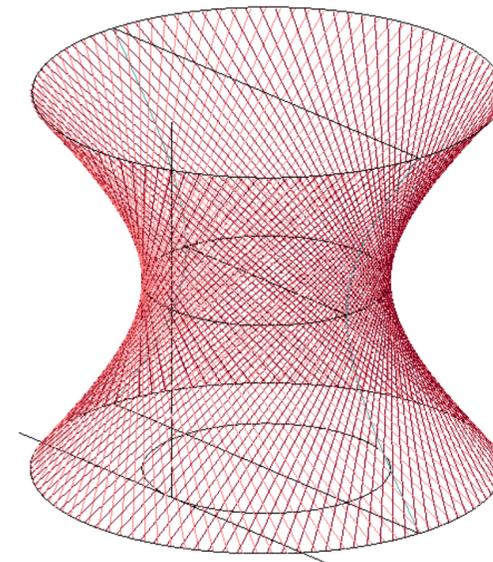
Marcação da geratriz (a vermelho).



Comando ARRAY (polar).



Comando ARRAY (polar) – separação das geratrizes sentido direita>esquerda e sentido esquerda>direita.



Hiperboloide de Revolução



Comandos SHADE e THICKEN.

# Exerc. 5.1 – Hiperboloide de Revolução

Para a determinação de várias superfícies recorreu-se aos comandos **EDGESURF** e **LOFT**.

1. Para a realização da parabolóide hiperbólica, construiu-se um paralelepípedo. Determinando as diagonais de cada face lateral, com uma **POLYLINE**.

De seguida, com o recurso ao comando **EDGESURF** (cria uma superfície, através de uma malha, recorrendo a 4 vértices ou curvas contíguas) criou-se a parabolóide hiperbólica.

2. De seguida, desenharam-se, sobre o plano x,y três circunferências (com raios diferentes e sem serem concêntricas). Utilizando o comando **MOVE**, deslocaram-se duas das circunferências, segundo o eixo z, de forma a não haver complanaridade entre quaisquer das circunferências.

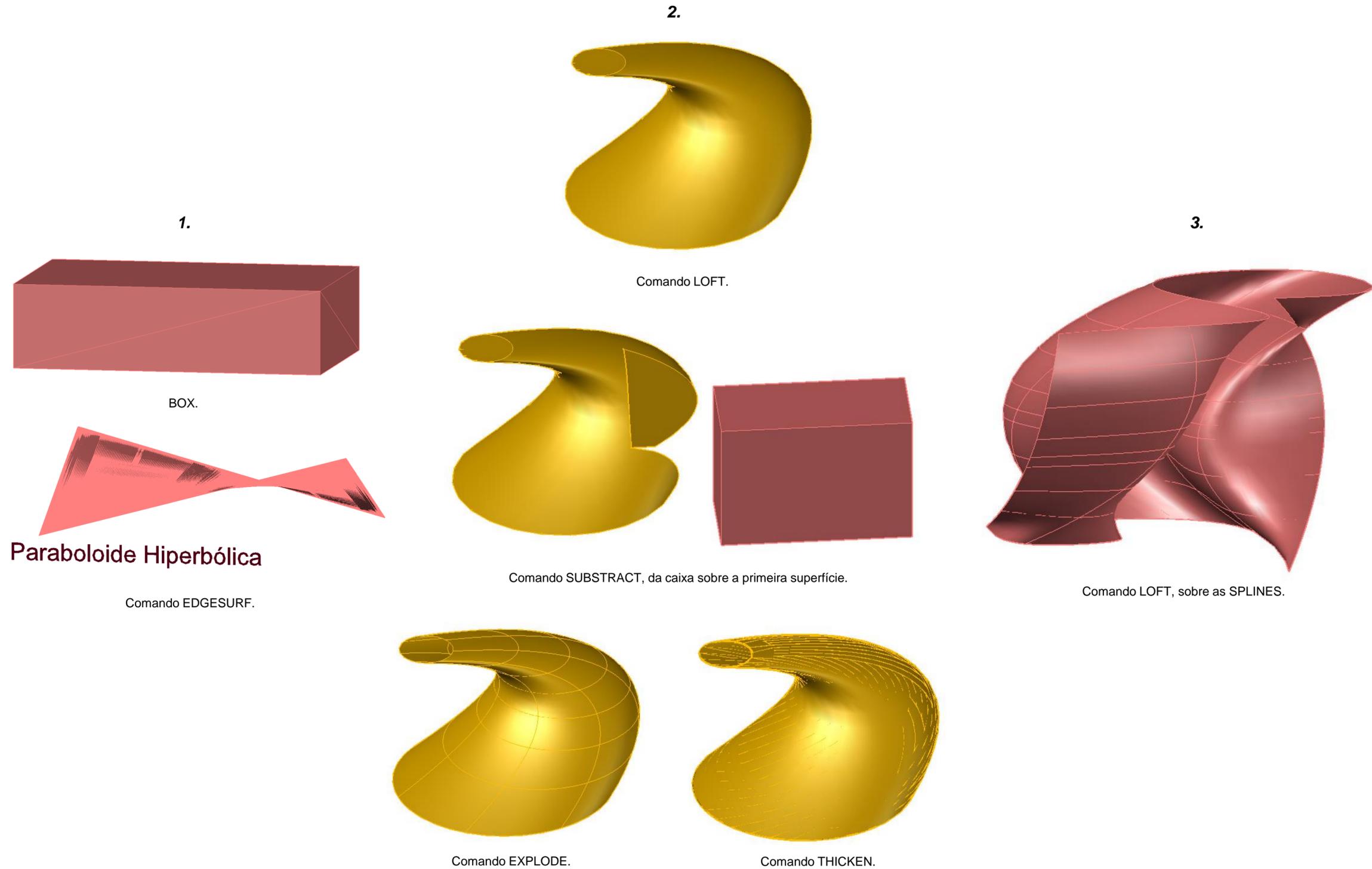
De seguida, com recurso ao comando **LOFT**, criou-se uma superfície (utilizando as circunferências ou quaisquer linhas, o comando cria superfícies ou sólidos).

Com o comando **SUBTRACT** (depois do desenho de uma qualquer caixa, com o comando **BOX**), verificou-se a subtração da caixa na superfície antes criada.

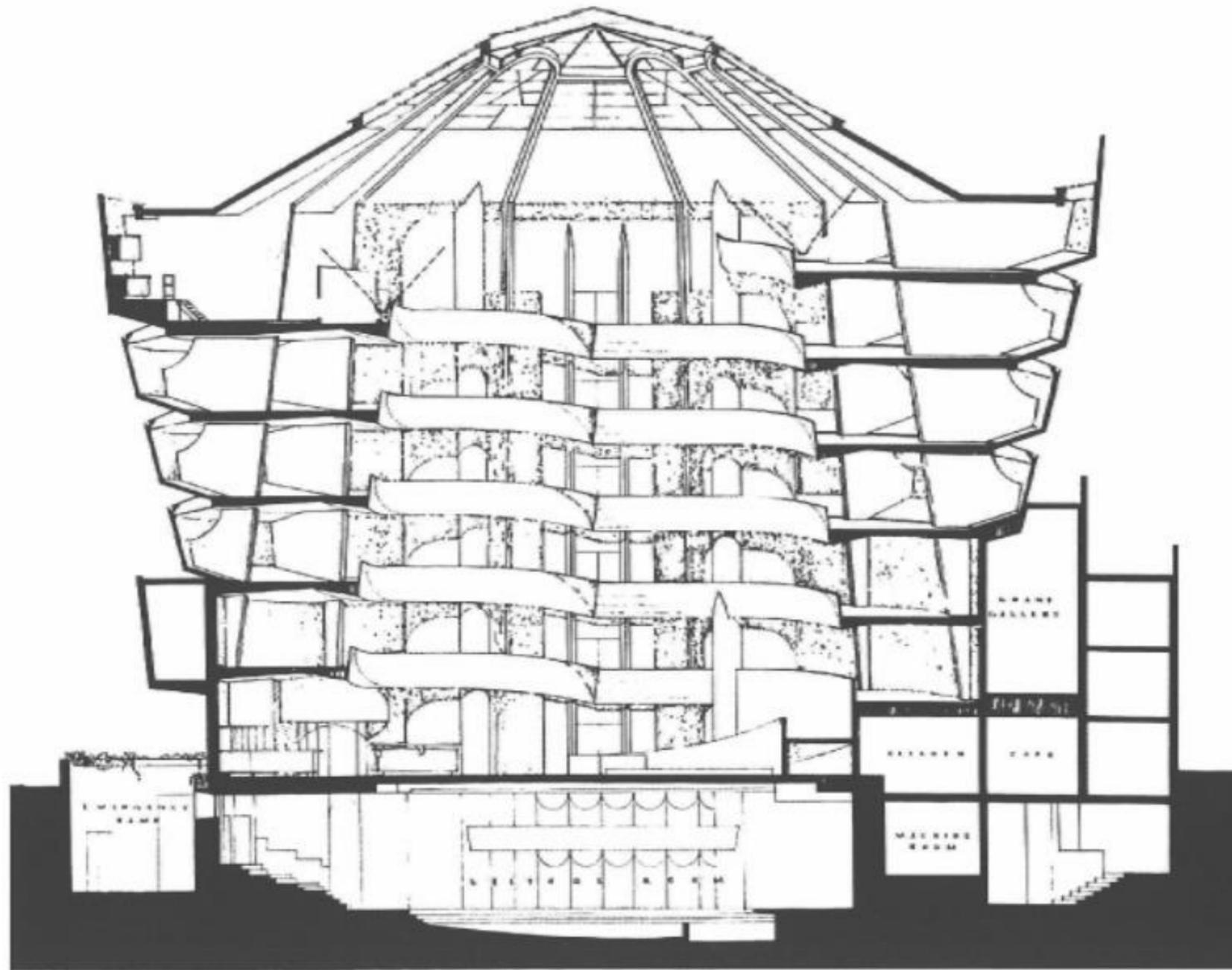
Explodindo a superfície original, com recurso ao comando **EXPLODE**, seguido do comando **MOVE**, é possível mover o topo da figura, percebendo que esta se trata de uma pele (e não de um sólido) que determina a superfície. Com recurso ao comando **THICKEN** atribui-se uma dada espessura à superfície: é importante considerar se a espessura deve ser *positiva* ou *negativa*, consoante o pretendido (uma vez que a espessura pode ser dada para fora ou para dentro da superfície, influenciando o resultado do sólido obtido).

3. De seguida, com o recurso ao comando **SPLINE**, desenharam-se três linhas curvas. Tendo manipulado a distância entre elas, segundo o eixo z (de forma às linhas não serem complanares), aplicou-se o comando **LOFT**.

Comandos principais: POLYLINE. EDGESURF. MOVE. LOFT. BOX. SUBTRACT. EXPLODE. THICKEN. SPLINE.



# Exerc. 5.2 – Comandos EDGESURF e LOFT



Exerc. 6 – Museu Guggenheim

O sexto exercício consistia na modelação do corpo principal do *Museu Guggenheim*, projeto do arquiteto *Frank Lloyd Wright*.

Partindo do desenho de um corte do Museu, *traça-se uma das secções da rampa, em corte*: o corpo central do museu é um cilindro irregular, vazado ao centro (uma rampa contínua percorre o vazio central – o percurso é feito com a disposição das obras de arte ao longo das rampas). Tendo a secção traçada, em **POLYLINE**, cria-se uma **HELIX**.

Uma vez que o tronco central vai alargando, conforme o museu sobe em altura, o comando **HELIX** tem, também, assim, de ser definido: estabelecendo um raio de uma primeira circunferência (da *circunferência inferior*), que *corresponde ao raio do vazio medido a uma cota mais baixa*, e uma segunda medida, de *raio maior* (*corresponde à distância à cota mais alta*): uma vez que em corte é possível distinguir *4 secções empilhadas* define-se que a hélix irá percorrer *4 voltas*.

Tendo a hélix desenhada, é necessário alinhar o início da mesma com um ponto da secção, com recurso ao comando **ALIGN**: a hélix será alinhada ao ponto de referência da *circunferência menor*, uma vez que a secção irá ser extrudida sobre esta. Com o comando **3DROTATE**, rebate-se a secção sobre um *plano frontal x,z*.

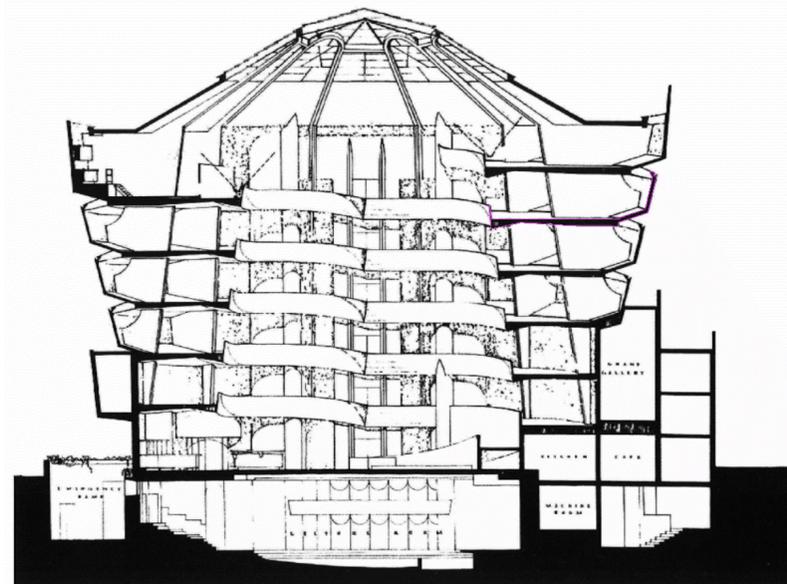
Com o comando **EXTRUDE**, a secção é extrudida, definindo como *referencial de extrusão a hélix*. O corpo central do museu é, então, modelado.

Como segunda parte do exercício, decidiu-se que este corpo poderia ser cortado, de forma a revelar, em corte, o seu interior.

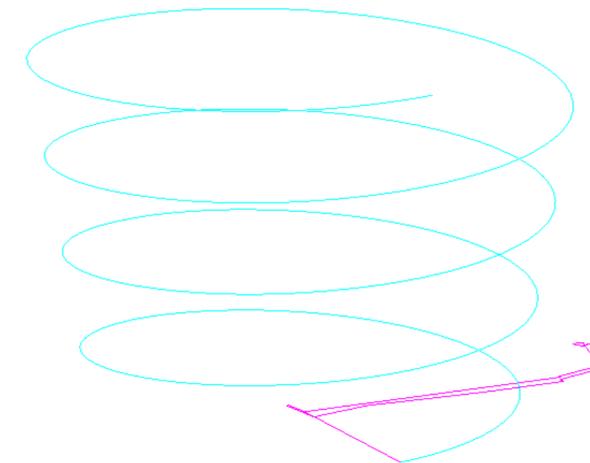
Desta forma, sobre o corpo do museu, assente numa base (**BOX**), desenha-se uma caixa.

Com recurso ao comando **SUBTRACT**, a parte do corpo do museu engolida pela caixa vai ser subtraída ao todo – revelando as rampas interiores do museu (tal como o desenho inicial, de referência).

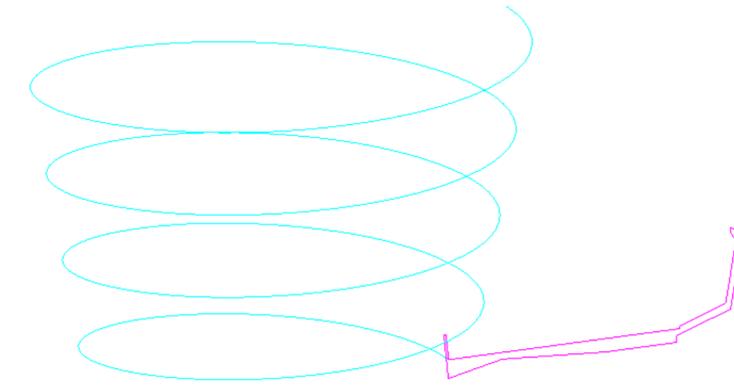
Comandos principais: POLYLINE. HELIX. ALIGN. 3DROTATE. EXTRUDE. BOX. SUBTRACT.



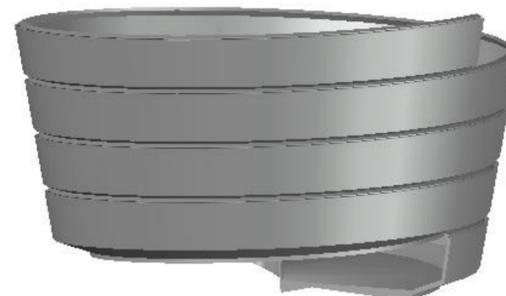
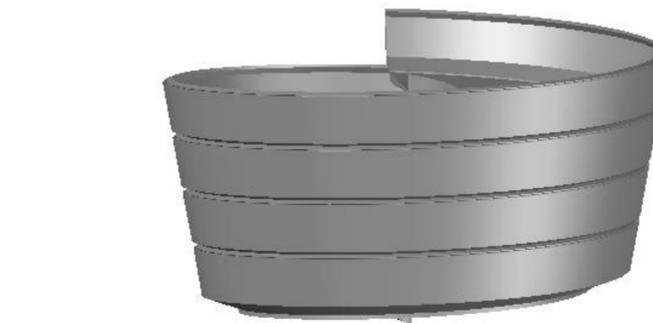
Corte/Secção do Museu.



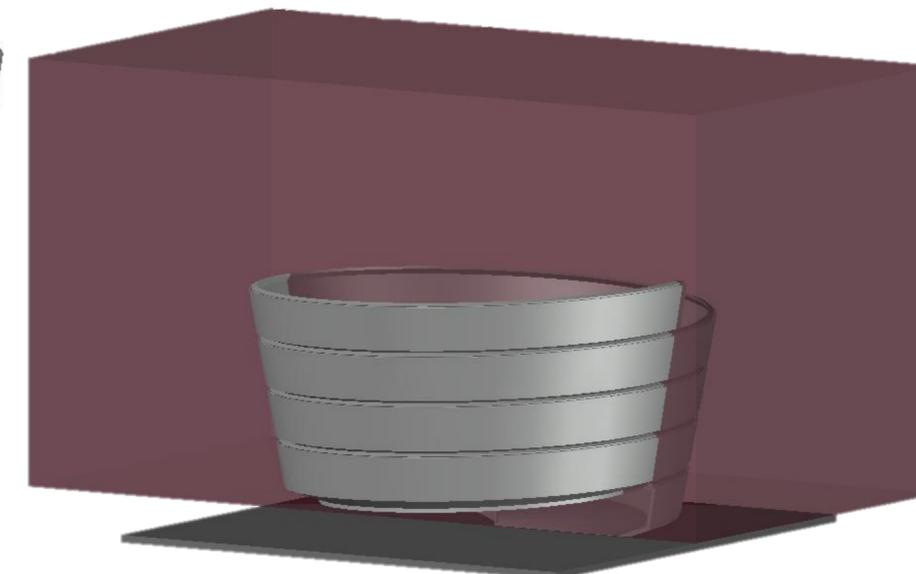
Comando HELIX.



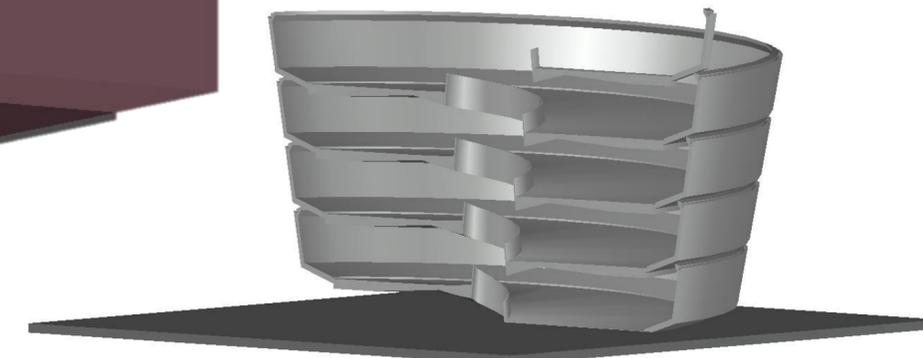
Alinhamento do começo da hélix com um ponto da secção.  
Comando 3DROTATE.



Comando EXTRUDE.



Desenho de uma caixa – comando BOX.



Comando SUBTRACT – subtração da caixa ao todo,  
revelando a vista interior do Museu.

# Exerc. 6 – Museu Guggenheim



Exerc. 7 – Lamparina

Este exercício marca a iniciação a um novo programa: 3DS MAX.

O programa 3DS MAX, da AUTODESK, consiste num programa de modelação tridimensional e de renderização e processamento de imagens e vídeos. É possível a importação de diferentes ficheiros (nomeadamente de ficheiros .dwg, importados do AUTOCAD), que podem, assim, ser modelados (ou, se já estiverem modelados, serem atribuídas diferentes texturas e materiais à modelação). O programa trabalha com a apresentação simultânea de diferentes vistas, sobre um mesmo objeto.

Uma vez que se trata de um programa novo, a demonstração deste primeiro exercício contém capturas de ecrã (que mostram os diferentes passos e informações adicionais).

O exercício consistia no desenho de uma *lamparina*, *mesa de apoio*, e *cenário envolvente* (paredes e suporte de pavimento).

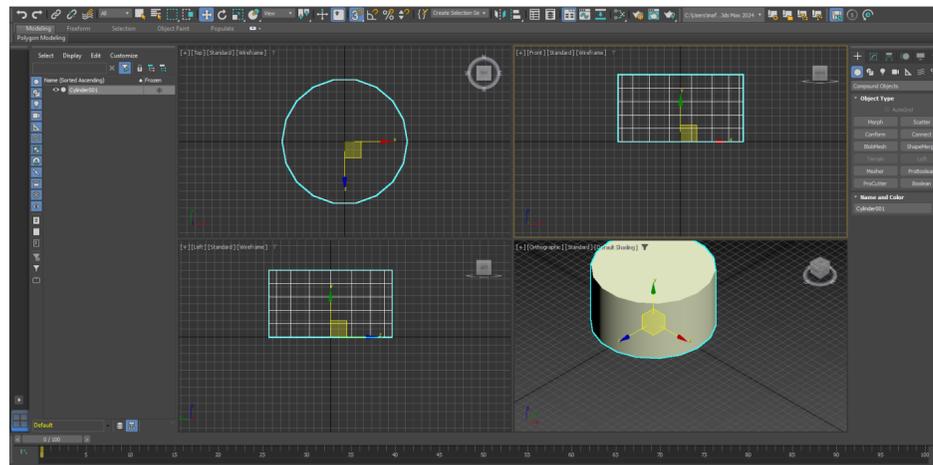
Desta forma, começou-se por desenhar a lamparina: para o tronco, desenhou-se um *CILINDRO* (um cilindro qualquer, ao qual, depois são atribuídas as dimensões pretendidas). De seguida desenha-se um *TORUS* (o processo é igual ao do desenho do cilindro). Movendo-o sobre o eixo z, estabelece-se o *torus*, ligeiramente acima do inicial.

Com recurso aos **COMPOUND OBJECTS**, subtrai-se o *torus* ao *cilindro* (criando uma reentrância no corpo da lamparina).

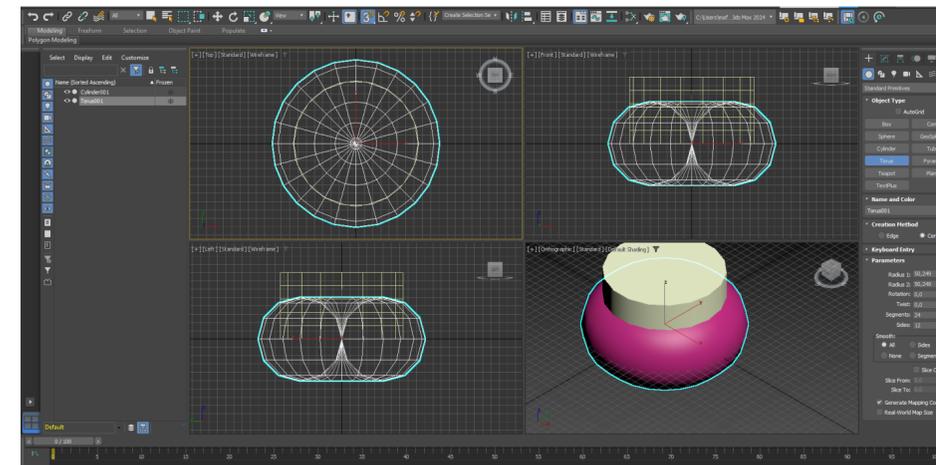
Modificando as propriedades do objeto, é possível determinar o seu contorno – *TAPER* (neste caso, alterando os valores iniciais, pretende-se arredondar o topo do cilindro).

De seguida, desenha-se o “botão” da lamparina – dois cilindros, que se interseam (mais uma vez, manipulam-se as diferentes dimensões dos cilindros, de forma a se obter a forma pretendida). Resta mover o botão para o exterior do corpo da lamparina.

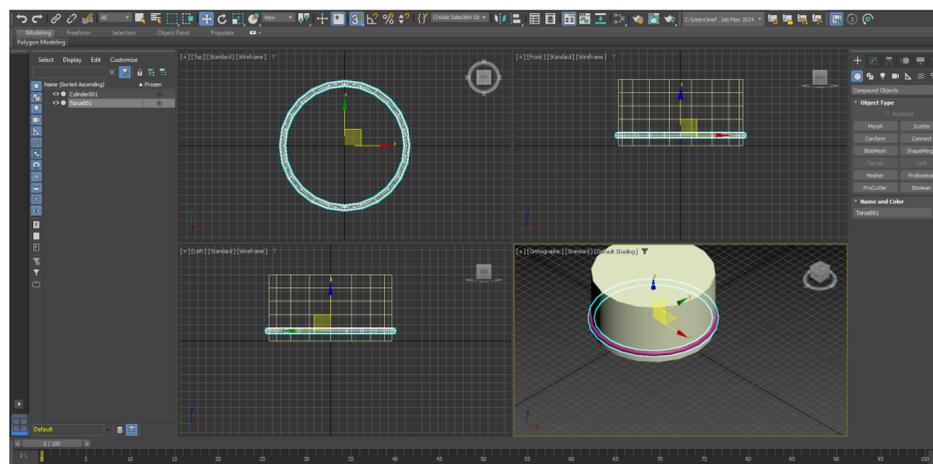
Comandos principais: CYLINDER. TORUS. BOOLEAN – TAPER. MOVE.



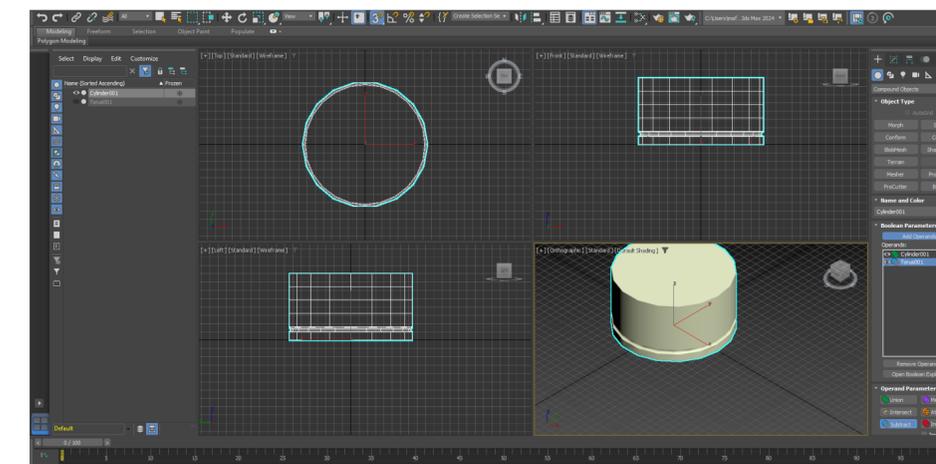
1. Cilindro Inicial.



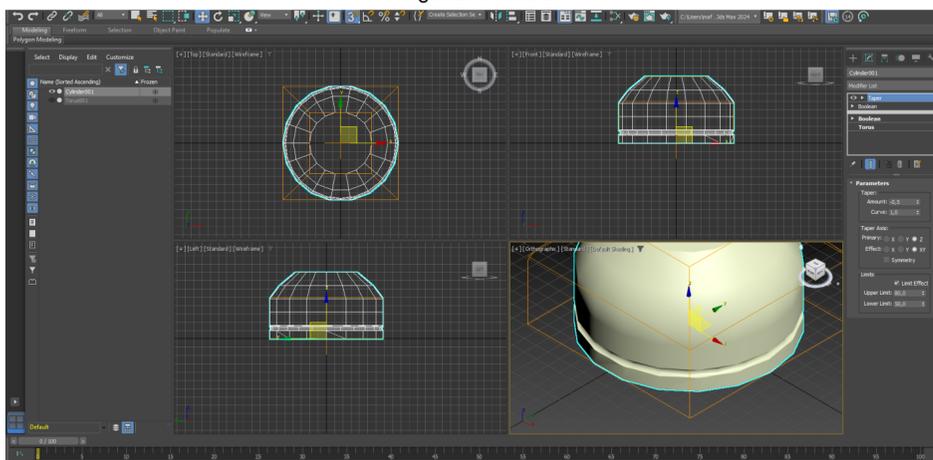
2. Torus.



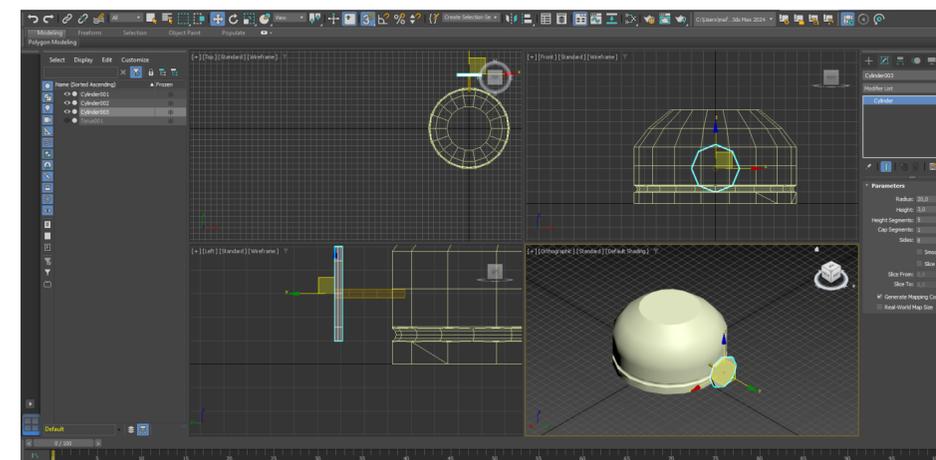
3. Ajuste das dimensões do Torus. Re-localização do Torus – segundo o eixo z.



4. Subtract.

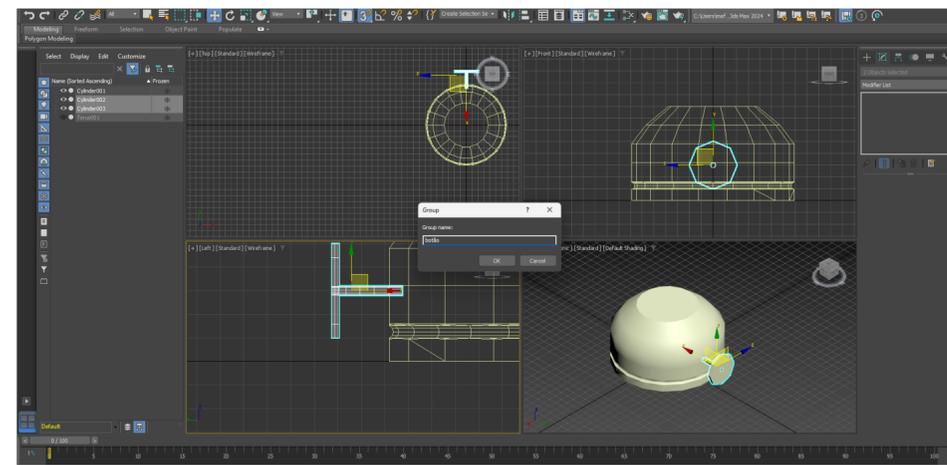


5. Boolean – TAPER.

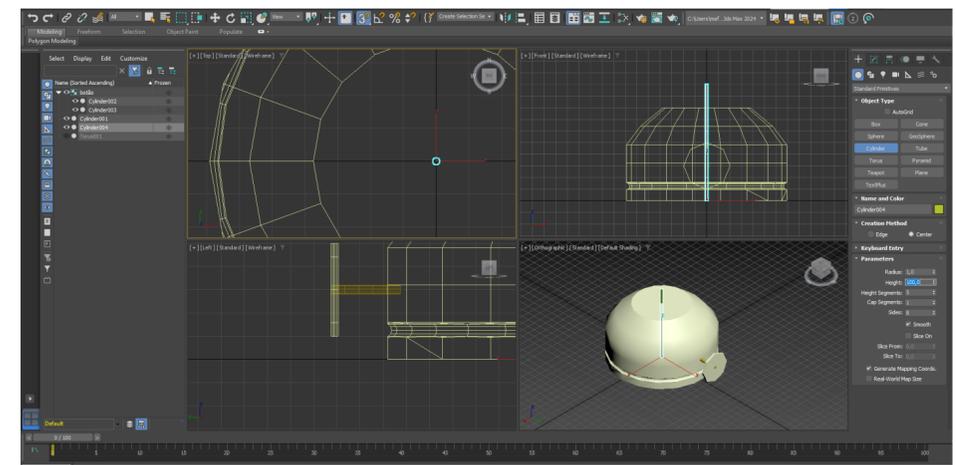


6. Desenho do “botão”. Re-localização do botão – segundo o eixo y.

# Exerc. 7 – Lamparina



7. Deslocamento do "botão".

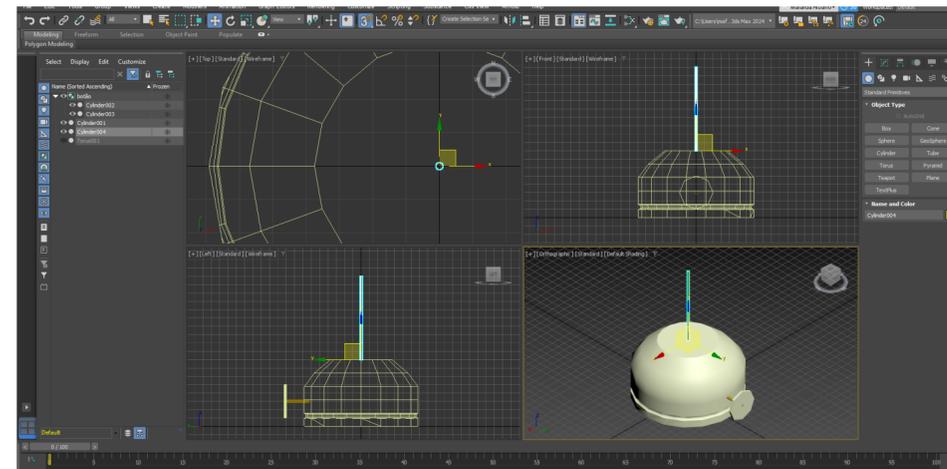


8. Cilindro da Chama.

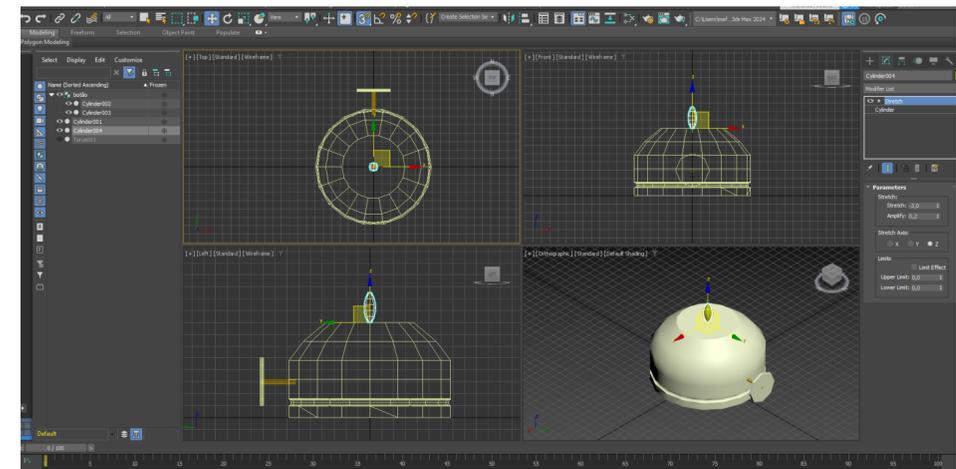
Para a sugestão de uma chama, começa-se por se desenhar um cilindro. Movendo o cilindro sobre o eixo z, determina-se a posição final da chama – no topo do corpo da lamparina. Ao alterar as dimensões do cilindro, e aplicando-lhe um **STRETCH**, a chama ganha uma forma amendoada. Acrescentando **NOISE** à forma – e alterando a **SCALE** e **SEEDS**, refratando o barulho da forma – a forma começa a assemelhar-se a uma chama.

Resta desenhar a campânula da lamparina. Para isto, com recurso a uma *vista frontal*, desenhou-se uma linha.

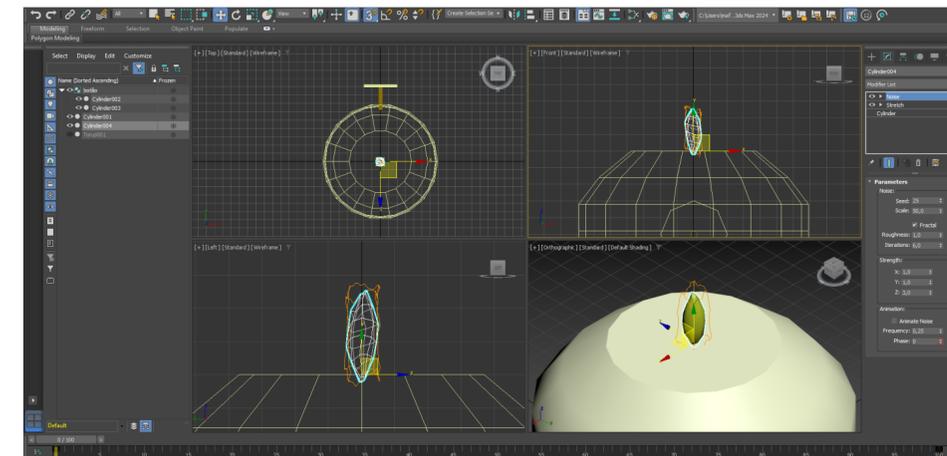
Comandos principais: CYLINDER. STRETCH. NOISE. LINE.



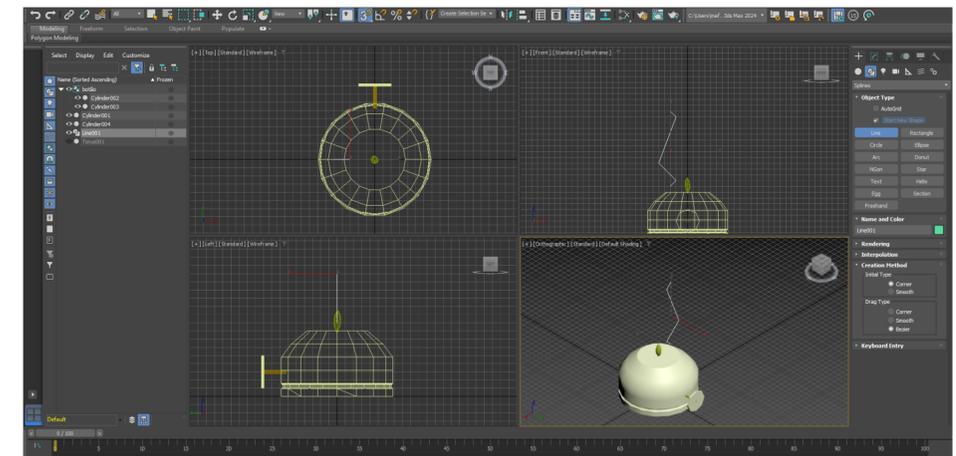
9. Deslocamento do cilindro, segundo o eixo z.



10. STRETCH do cilindro da chama.



11. Aplicação de NOISE. Experimentação com os valores de SCALE e SEED.



12. Desenho da linha do vidro da lamparina.

# Exerc. 7 – Lamparina

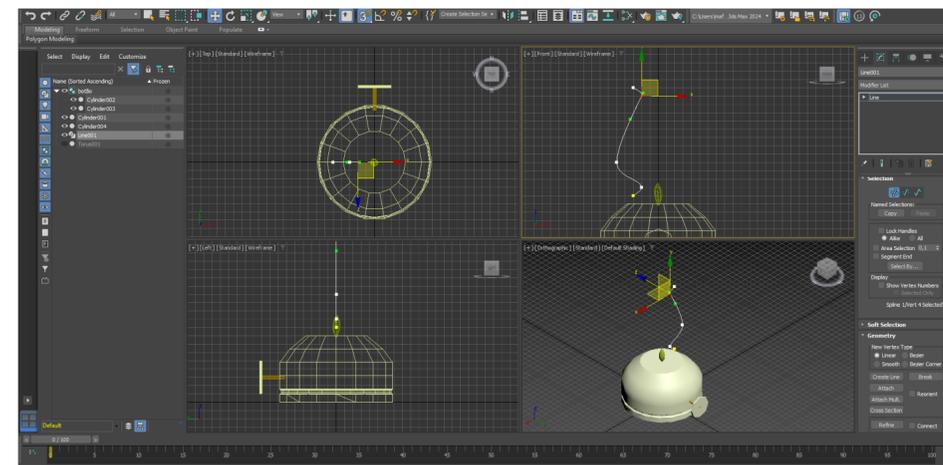
Com o botão direito do rato é possível alterar as propriedades dos vértices – neste caso, determinou-se que os vértices deveriam ser arredondados, não formando arestas vincadas. Com a linha selecionada, seleciona-se o comando **LATHE – AXIS**, e cria-se uma superfície axial, da campânula.

Tendo a lamparina desenhada, resta desenharmos as peças de mobiliário. Com dimensões pré-estabelecidas, começa-se por se desenharmos uma **BOX** (o tampo da mesa). Com o mesmo comando, desenha-se a primeira perna da mesa. Com recurso às coordenadas do seu eixo, é possível manipular o seu posicionamento, em relação ao tampo da mesa. O comando **MIRROR** permite espelhar a primeira perna da mesa, restando definir as suas coordenadas exatas. Era requerido que a lamparina não ficasse ao centro da mesa – mas que fosse puxada para a direita (de forma a que, quando aplicados os materiais, textura e luz, a luz projetada e, conseqüentemente, as sombras, não estivessem ao centro). Para deslocar a lamparina, utiliza-se o comando **MOVE**.

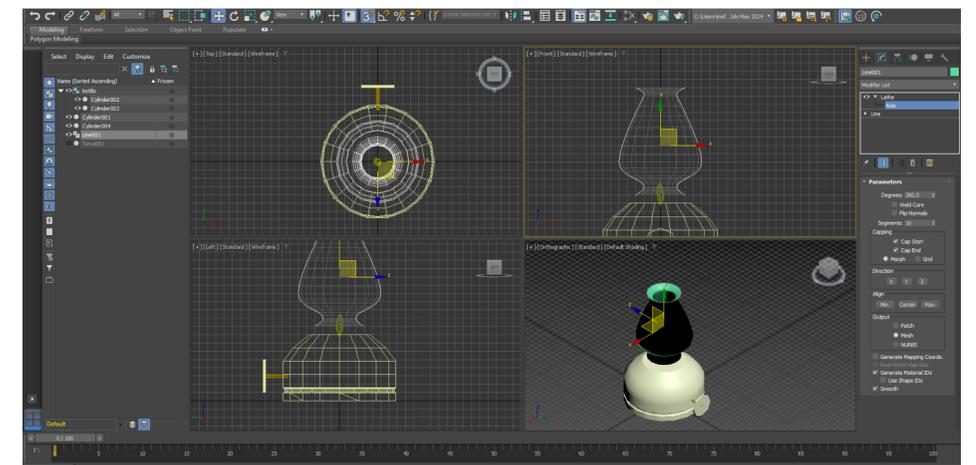
É de notar que, desligando os **SNAPS TOGGLE**, o elemento selecionado, que se pretende mover, apenas se desloca segundo os eixos x, y e z.

Para a modelação das paredes e pavimento utiliza-se, novamente, o comando **BOX**, definindo, posteriormente, as coordenadas pretendidas.

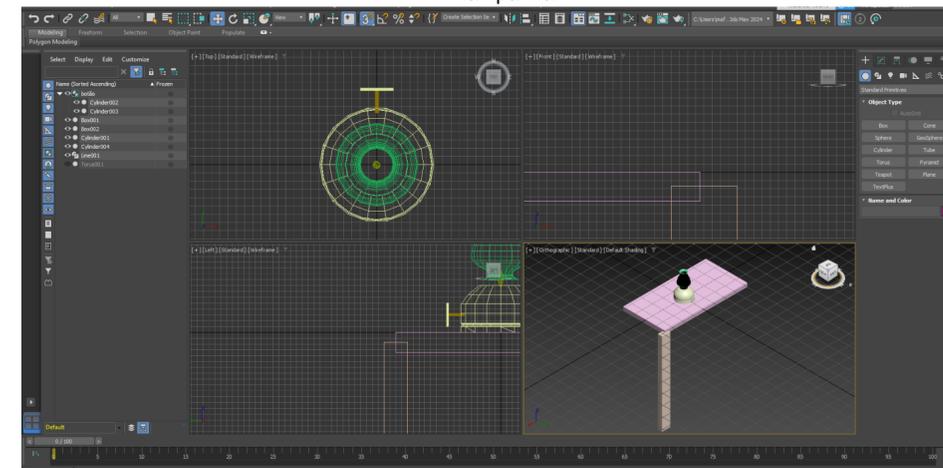
Comandos principais: LATHE – AXIS. MIRROR. MOVE. BOX



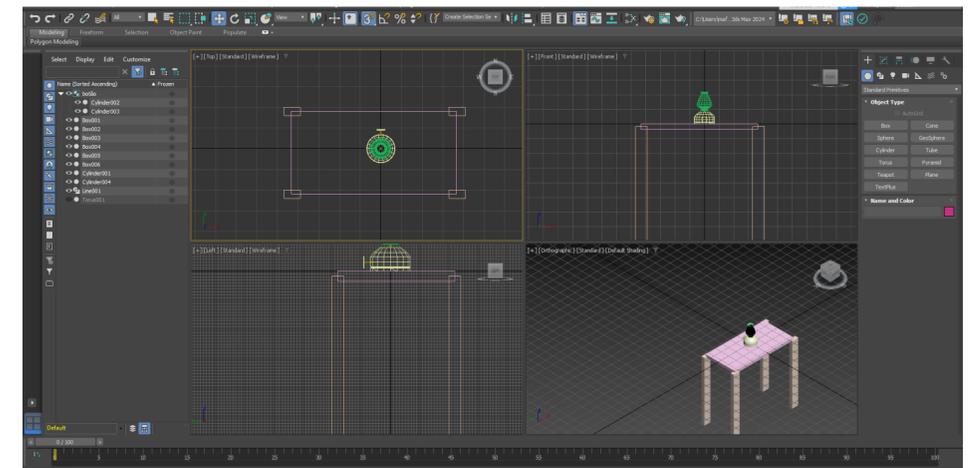
10. Determinação das propriedades da linha do vidro da lamparina..



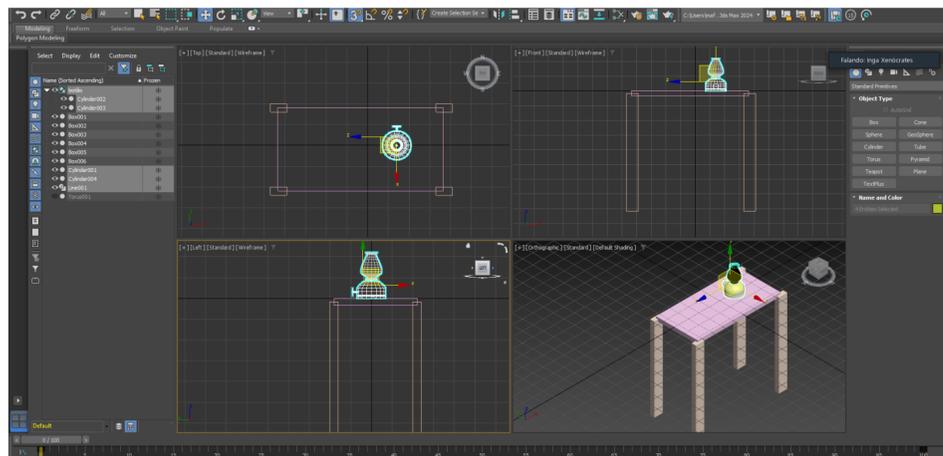
11. Comando LATHE - AXIS.



12. Comando BOX e ajuste das coordenadas.



13. Comando MIRROR.



5. Comando MOVE..



6. Modelação final..

# Exerc. 7 – Lamparina

A segunda parte deste trabalho consistia na atribuição de materiais e texturas à modelação.

Para isto, exploraram-se duas abordagens possíveis:

- A primeira consiste na criação (e posterior aplicação) de um material feito a partir do 0.

O comando **MATERIAL EDITOR** permite a criação de amostras de materiais, com características específicas – permitindo manipular a matiz, tonalidade, brilho e textura. A cada novo material é atribuída uma tela em branco.

Assim sendo, escolhendo uma tela em branco, procurou replicar-se o latão. Tendo escolhido uma cor semelhante à do latão (algo entre o amarelo e o dourado), manipularam-se as diferentes variantes disponíveis, de forma a puder aplicar o novo material no corpo da lamparina.

- Para uma segunda abordagem, procuraram-se imagens do material pretendido, na Internet, atribuindo-as à tela em branco (este é o método utilizado em todos os elementos da modelação, com exceção da campânula, uma vez que garante maior veracidade na réplica dos materiais).

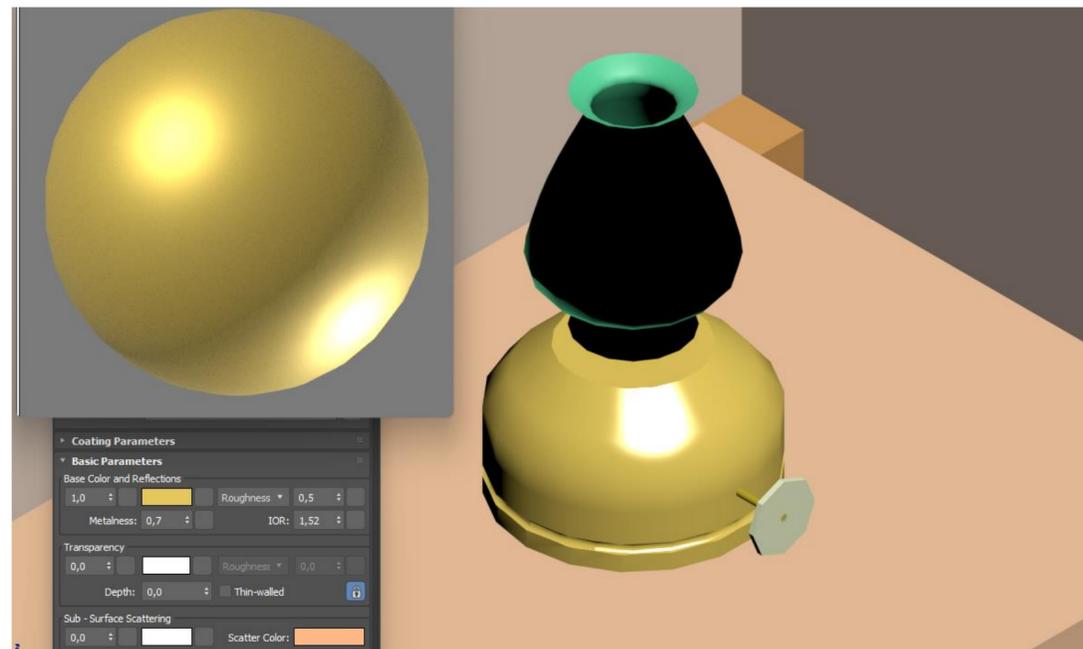
Desta forma, procurou-se uma imagem de latão escovado (uma vez que, por tem mais variações e textura, torna a modelação mais realista). Tendo escolhida a imagem pretendida, criou-se um **BITMAP** (Generic Map > Base Color > General > Bitmap): consiste numa estampa da imagem, sobre a qual é possível manipular as predefinições (manipular a escala da imagem – vertical e horizontal – rotação ou espelhá-la).

Selecionando o material criado, resta aplicá-lo no corpo pretendido.

Para a criar a campânula da lamparina, uma vez que é feita de vidro, fazia sentido replicar o material sem recurso a imagens (uma vez que o processo é rápido e expedito). Assim, seleccionou-se uma matiz azulada, com uma alta percentagem de transparência.

Para a replicar a chama, criou-se um BITMAP, com uma imagem de chamas. Tendo as definições da imagem base definidas, aplicou-se um filtro alaranjado sobre a amostra, de forma a criar uma amostra uniforme e credível.

Comandos principais: MATERIAL EDITOR.



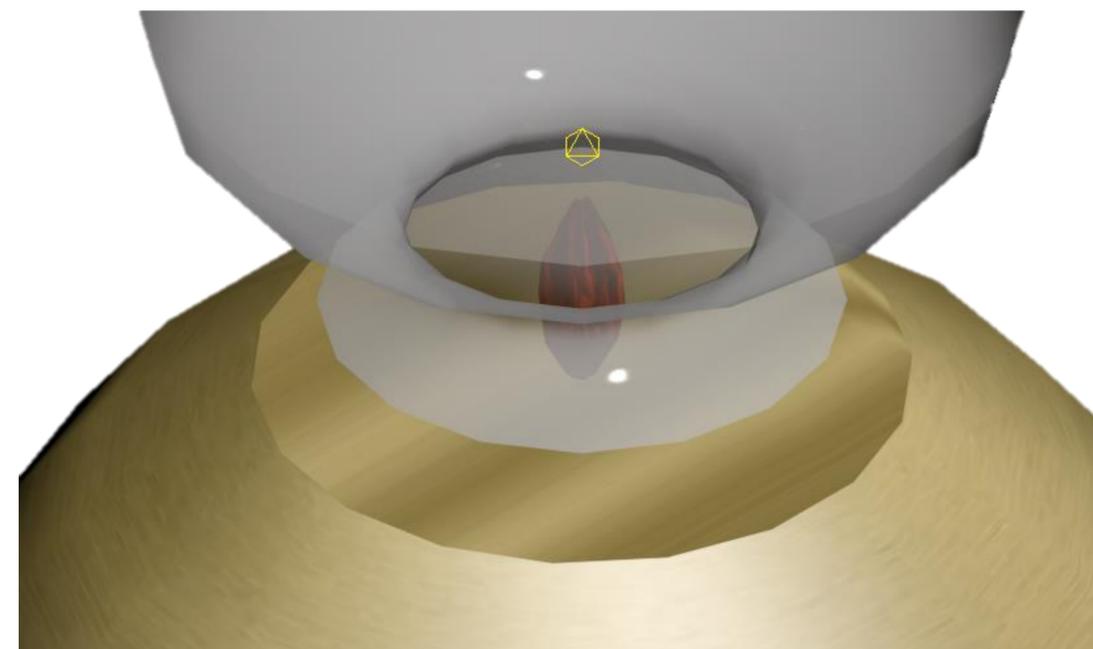
Criação de latão, a partir do 0 – manipulação das diferentes variantes.



Criação de latão, utilizando uma imagem para a amostra.



Aplicação do vidro, na campânula.



Chama da lamparina.

# Exerc. 7.1 – Materiais e Texturas

Para a mesa e cenário envolvente, foi mais intuitivo (e permitia resultados mais realistas) escolher imagens e aplicá-las às telas.

Tendo uma imagem escolhida, para o tampo da mesa (e tendo manipulado as suas propriedades), utilizou-se o **PHOTOSHOP** para aumentar o contraste e exposição da imagem, de forma a que a imagem aplicada nas pernas da mesa permitisse que estas se diferenciassem do tampo.

Optou-se por um pavimento em tabuado de madeira, com uma aparência mais gasta (*a intenção foi criar ambiência crua, contrastante e intimista, na cena*).

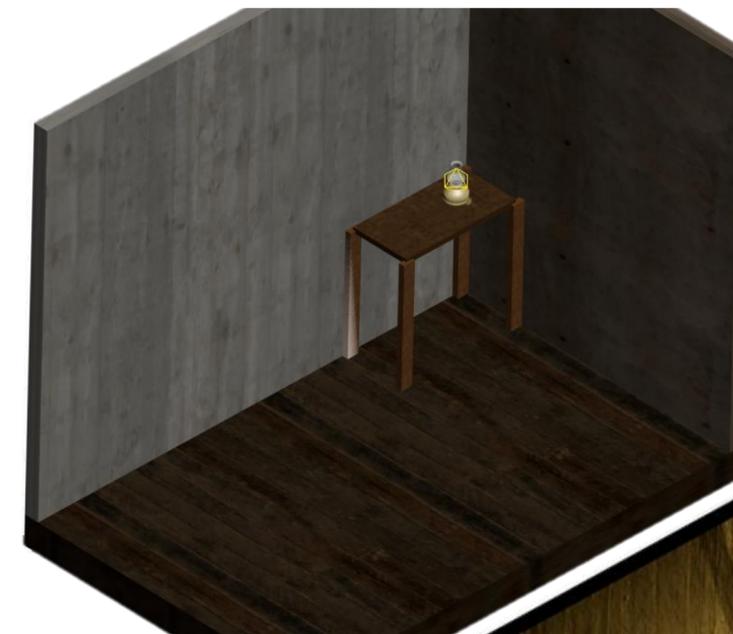
Para as paredes, de betão, exploraram-se dois acabamentos diferentes: numa, o *tabuado da cofragem* marca um ritmo vertical, na outra, o um *acabamento mais irregular e rugoso*, sem um ritmo marcado.

De forma a criar a rugosidade característica do betão e madeira, para a mesa e cenário, criaram-se **BUMP MAPS** (*Special Maps > Bump Map*): à semelhança dos mapas criados para a aplicação das imagens nos corpos, o **BUMP MAP** duplica essa mesma imagem (*definindo as mesmas propriedades estabelecidas no anterior*) simulando a profundidade das texturas (*facilita a renderização, uma vez que simula uma geometria a partir das duas dimensões*).

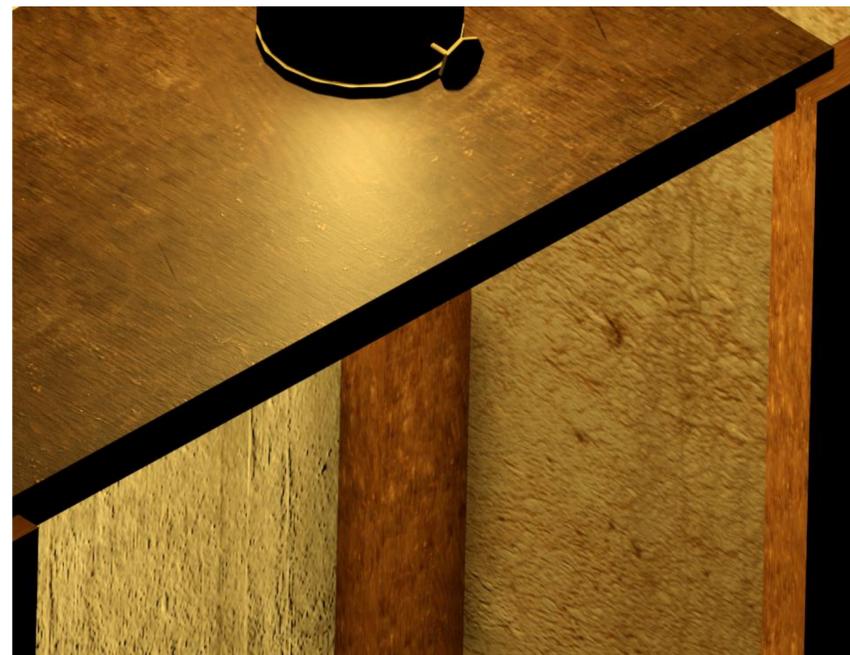
Para a inserção de um *foco de luz* (simulando a luz da chama), utilizou-se o comando **CREATE – lights**. Escolhendo uma *luz Standard* (do tipo *OMNI*), restava localizar o foco nos planos (após alguma experimentação, constatou-se que o melhor resultado advinha da sua colocação ligeiramente acima da chama). Faltava definir a *intensidade, matiz e tonalidade* pretendidas.

Com a luz e sombras ativadas, o exercício fica concluído.

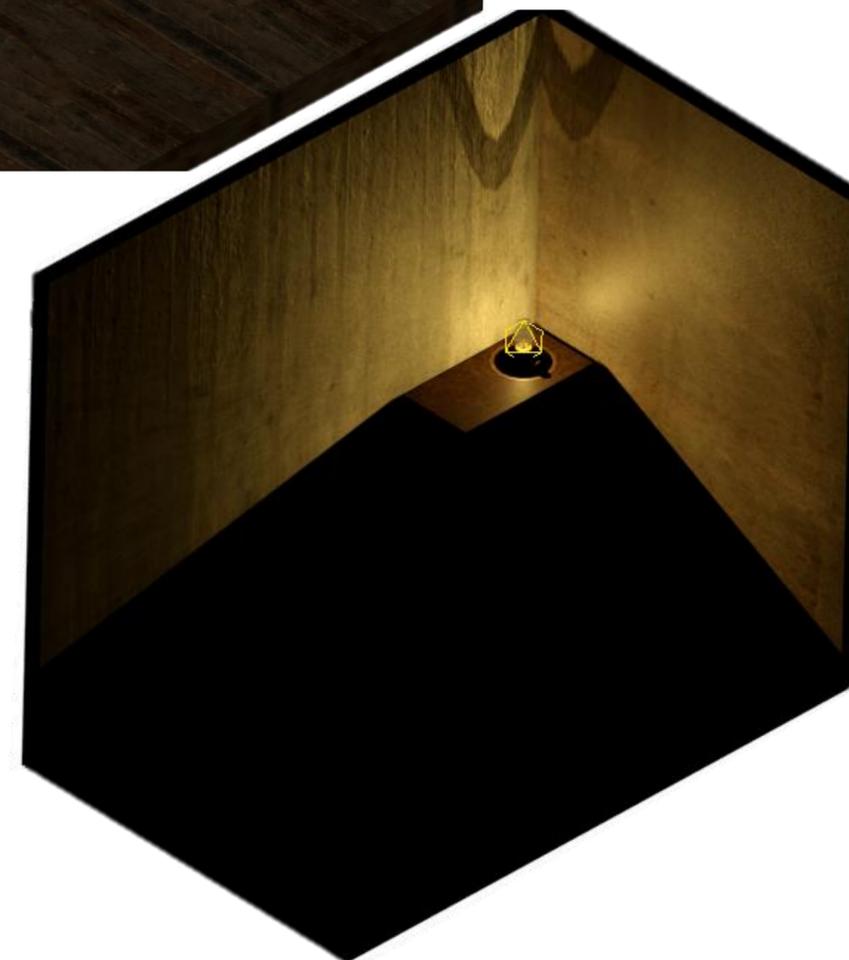
Comandos principais: MATERIAL EDITOR. CREATE – LIGHTS.



Aplicação do betão e madeira.



BUMP MAP + aplicação de um foco de luz.



Resultado final.

# Exerc. 7.1 – Materiais e Texturas