

CURL

Rubén Díaz Zapico - uo283204

Enrique Hilanderas Corbillón - uo282867

Diego García García - uo282492

Índice

- Introducción
- Historia
- Decisiones arquitectónicas
- Conclusiones

Introducción



```
computer@computer: ~/curl
computer@computer:~/curl$ curl https://www.geeksforgeeks.org >log.html --libcurl code.c
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 94358  0 94358  0    0  247k    0 --:--:-- --:--:-- --:--:-- 247k
computer@computer:~/curl$ head -n 30 code.c
/***** Sample code generated by the curl command line tool *****/
* All curl_easy_setopt() options are documented at:
* http://curl.haxx.se/libcurl/c/curl_easy_setopt.html
*****/
#include <curl/curl.h>

int main(int argc, char *argv[])
{
    CURLcode ret;
    CURL *hnd;

    hnd = curl_easy_init();
    curl_easy_setopt(hnd, CURLOPT_URL, "https://www.geeksforgeeks.org");
    curl_easy_setopt(hnd, CURLOPT_USERAGENT, "curl/7.47.0");
    curl_easy_setopt(hnd, CURLOPT_MAXREDIRS, 50L);
    curl_easy_setopt(hnd, CURLOPT_TCP_KEEPALIVE, 1L);

    /* Here is a list of options the curl code used that cannot get generated
    as source easily. You may select to either not use them or implement
    them yourself.

    CURLOPT_WRITEDATA set to a objectpointer
    CURLOPT_WRITEFUNCTION set to a functionpointer
    CURLOPT_READDATA set to a objectpointer
    CURLOPT_READFUNCTION set to a functionpointer
    CURLOPT_SEEKDATA set to a objectpointer
    CURLOPT_SEEKFUNCTION set to a functionpointer
    CURLOPT_ERRORBUFFER set to a objectpointer
    CURLOPT_STDERR set to a objectpointer
    CURLOPT_HEADERFUNCTION set to a functionpointer
computer@computer:~/curl$
```


Introducción

Código abierto



Fácil de usar con
muchas opciones



Instalado en
Windows y Linux



Introducción

¿Qué es **curl**://?

Es una herramienta de línea de comandos de código abierto que permite la transferencia de datos a través de varios protocolos de red, como HTTP, FTP, SMTP entre otros.



Historia

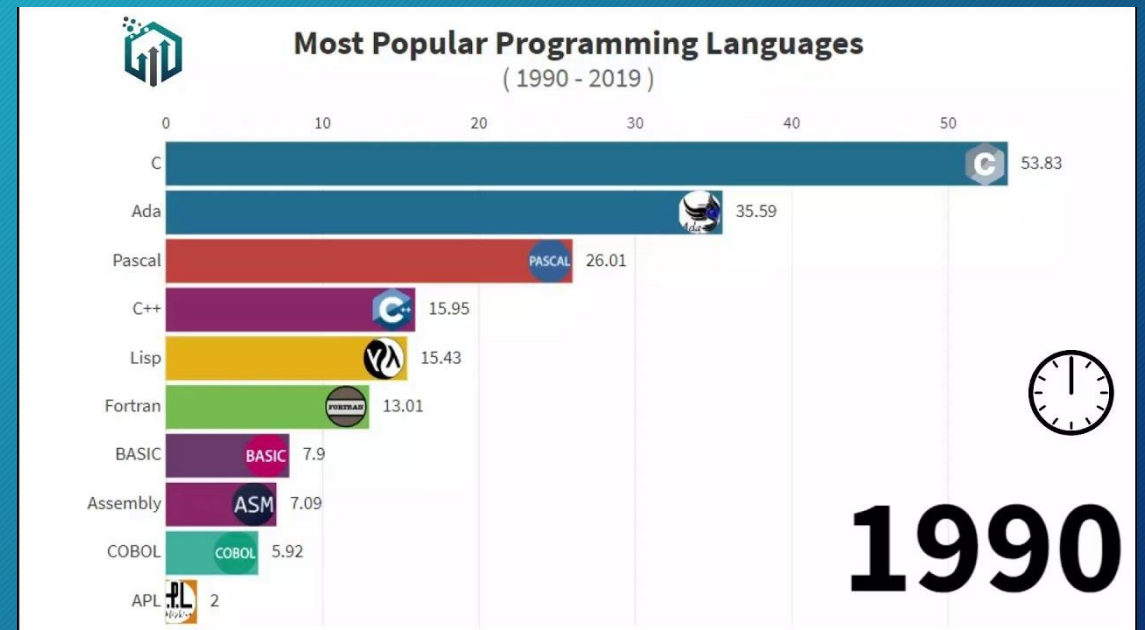
- HTTP GET
- Modificaciones
- cURL
- Popularidad



Decisiones arquitectónicas

Uso de como lenguaje de programación

- Era de los lenguajes más usados en la época.
- Además, permite ser usado en Linux y Windows.





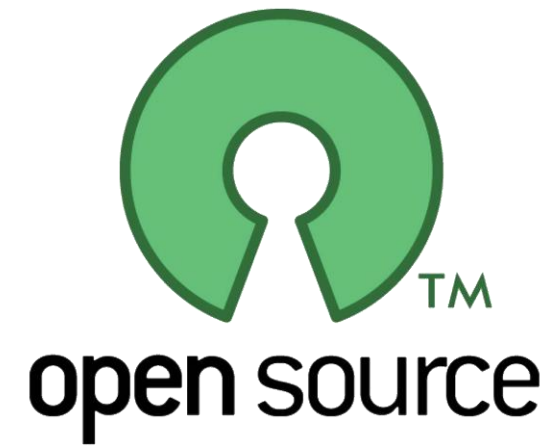
Decisiones arquitectónicas

Rust

Decisiones arquitectónicas



- Open Source, para contribuir mediante git.
- Librerías de terceros para manejar TLS, SSH.



Conclusiones



PENSAR EN EL FUTURO,
EN EL CAMBIO



APRENDER DE LOS
ERRORES



IMPACTO DECISIONES
ARQUITECTÓNICAS
(C -> RUST)



Conclusiones



Trabajo en equipo



Importancia de GIT



Buena
documentación

```
        operation == "MIRROR_X":  
            mirror_mod.use_x = True  
            mirror_mod.use_y = False  
            mirror_mod.use_z = False  
        operation == "MIRROR_Y":  
            mirror_mod.use_x = False  
            mirror_mod.use_y = True  
            mirror_mod.use_z = False  
        operation == "MIRROR_Z":  
            mirror_mod.use_x = False  
            mirror_mod.use_y = False  
            mirror_mod.use_z = True
```

```
        #selection at the end -add  
        mirror_ob.select= 1  
        mirror_ob.select=1  
        context.scene.objects.active  
        ("Selected" + str(modifier.name))  
        mirror_ob.select = 0  
        = bpy.context.selected_objects  
        data.objects[one.name].select
```

```
print("please select exactly
```

```
--- OPERATOR CLASSES ---
```

```
types.Operator):  
    X mirror to the selected  
    + mirror_mirror_x"
```

“Things are never done.
It’s doing things”