

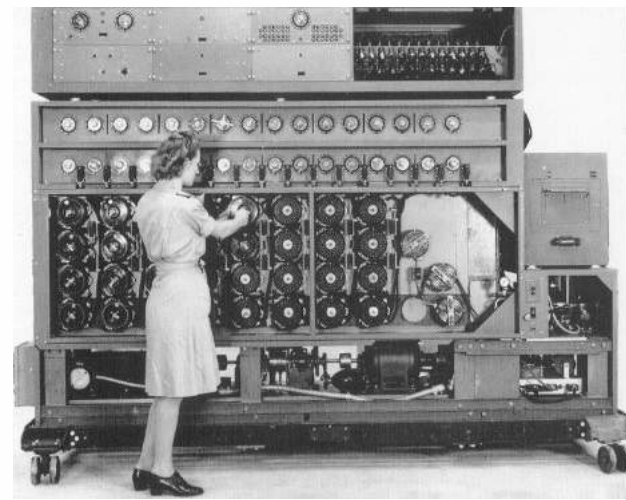
# Cohetería Computacional

## Clase general

- Historia y utilidad
- Objetivo de la sección
- Pre pre intro a programación
- Sección interactiva

# Historia y utilidad

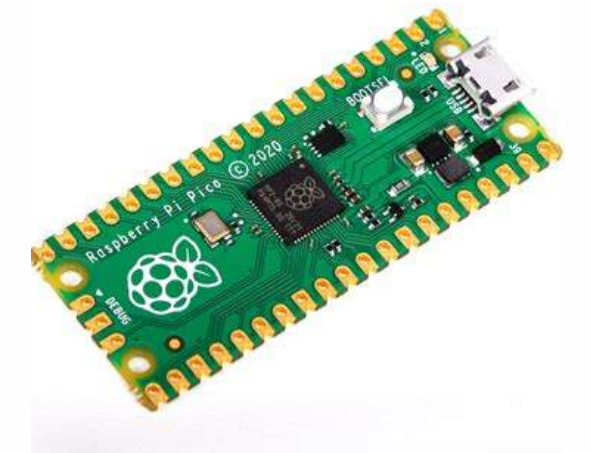
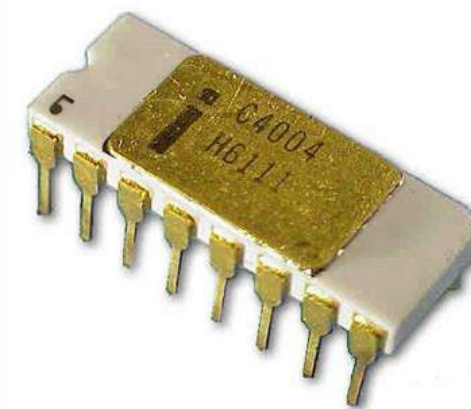
# Hardware



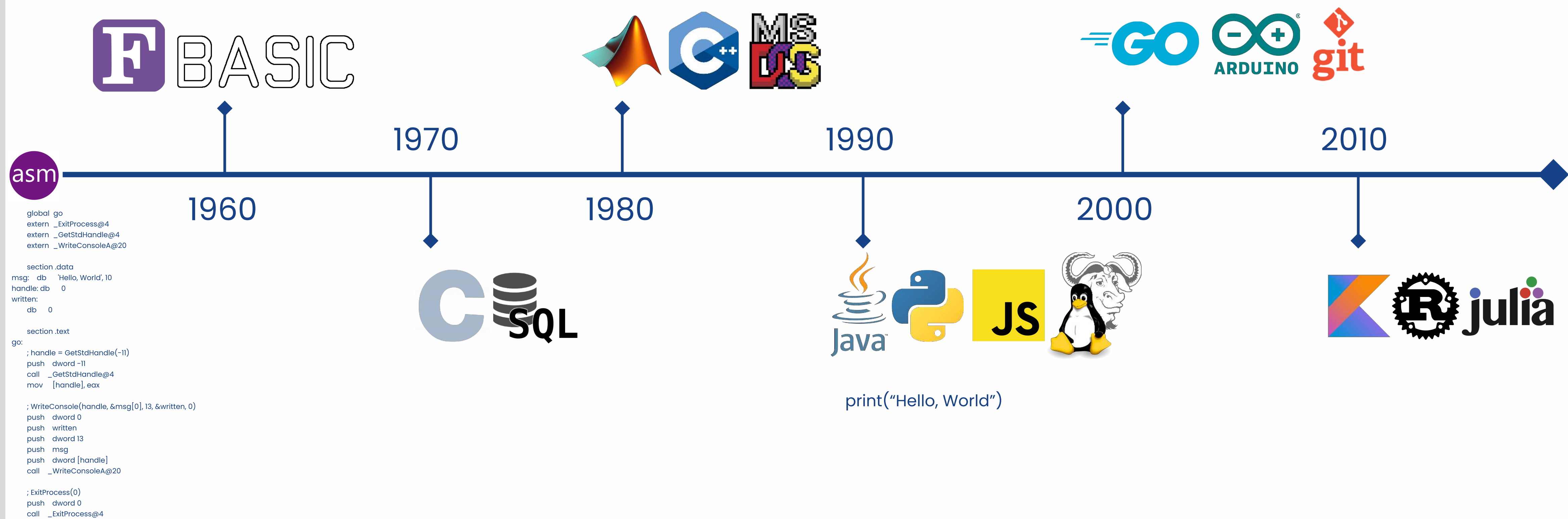
1970

1940

2000



# Software



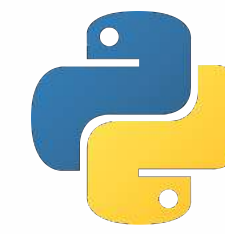


# Computación en la NASA

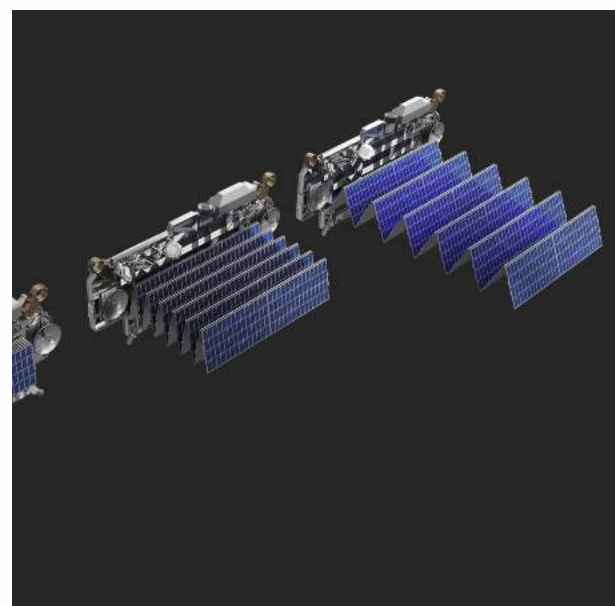


Gran trayectoria con múltiples tecnologías y áreas de trabajo muy variadas

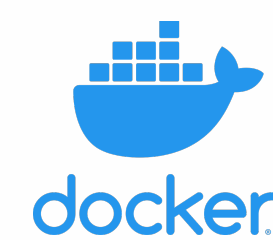
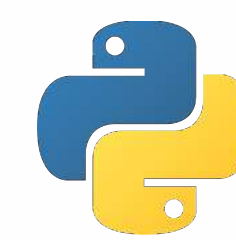
Proyectos open source en ***code.nasa.gov***



# Computación en SpaceX



Gran enfoque en experimentación e innovación y permite adoptar nuevas tecnologías

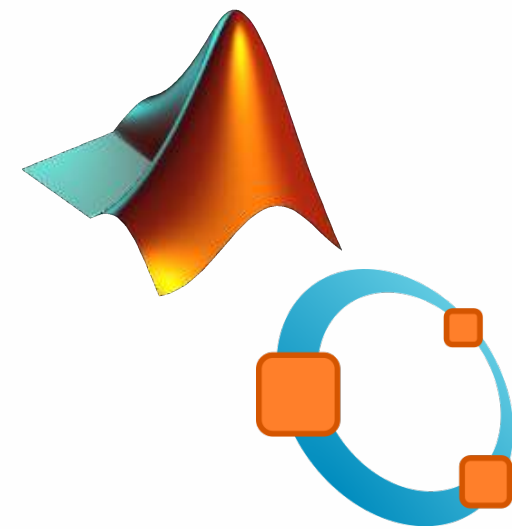


# Objetivo de la sección



# Colaborar entre equipos

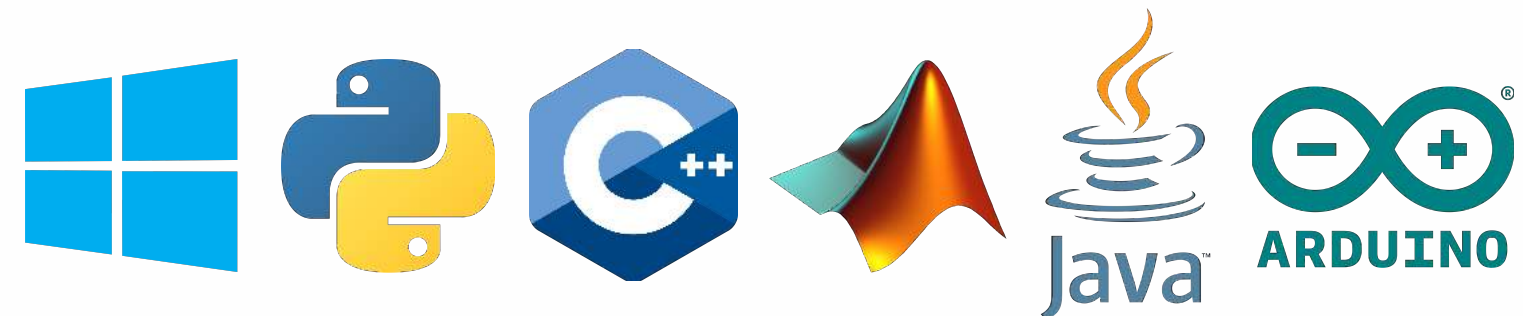
- Automatizar tareas repetitivas de otras áreas
- Ayudar a interpretar, traducir u optimizar algoritmos
- Contribuir con herramientas propias
- Facilitar la adopción de software libre



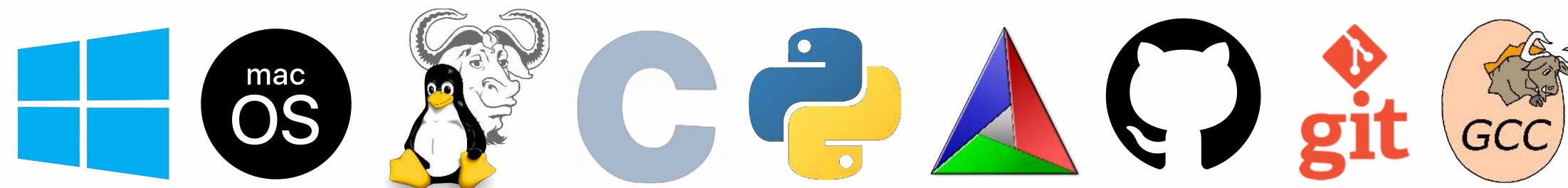


# Tecnologías usadas

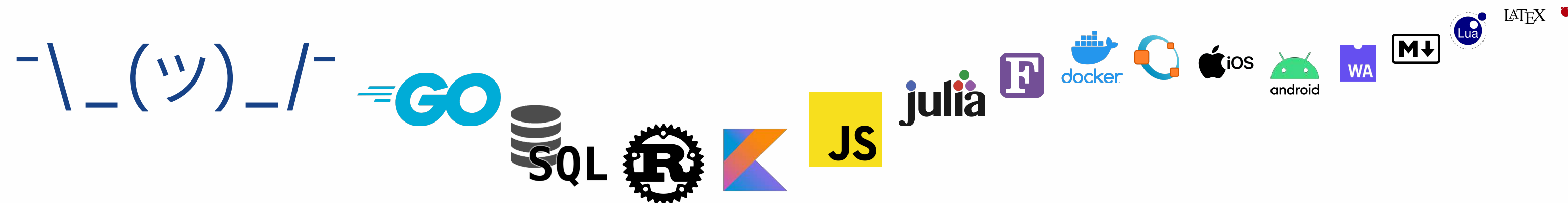
- Conocimiento común en integrantes



- Usado en proyectos actuales



- ¿Podemos usar \_\_\_\_\_ ?



# Pre pre intro a programación

# Tipos de variables

Las variables permiten nombrar información fácilmente

```
un_numero = 42
```

```
una_palabra = "hola"
```

En algunos lenguajes se debe especificar el tipo de variable

```
int un_numero = 42;
```

```
string una_palabra = "hola";
```

## Algunos tipos de variables que pueden ser útiles hoy (lenguaje C)

```
int numeroEntero = -216162;
```

```
float numeroReal = 68.000152;
```

```
double numeroRealPreciso = 68.00015234441;
```

Una vez se declaró el tipo de la variable, no se tiene que volver a declarar

```
numeroEntero = 16;
```

```
numeroReal = -6.0f;
```

```
numeroRealPreciso = 0.0;
```



# Funciones

Se pueden aislar secciones de código para reutilizar en distintos lugares y estas pueden retornar un resultado del tipo especificado

```
int sumarEnteros(int num1, int num2)
{
    return(num1+num2);
}
```

# Código estructurado

Se ordena las instrucciones mediante pruebas lógicas con los **if/else**

```
if (numero <= 0) {  
    terminar(-1);  
} else {  
    continuar();  
}
```

# Sección interactiva

# Simulación de Arduino en ***tinkercad.com***