

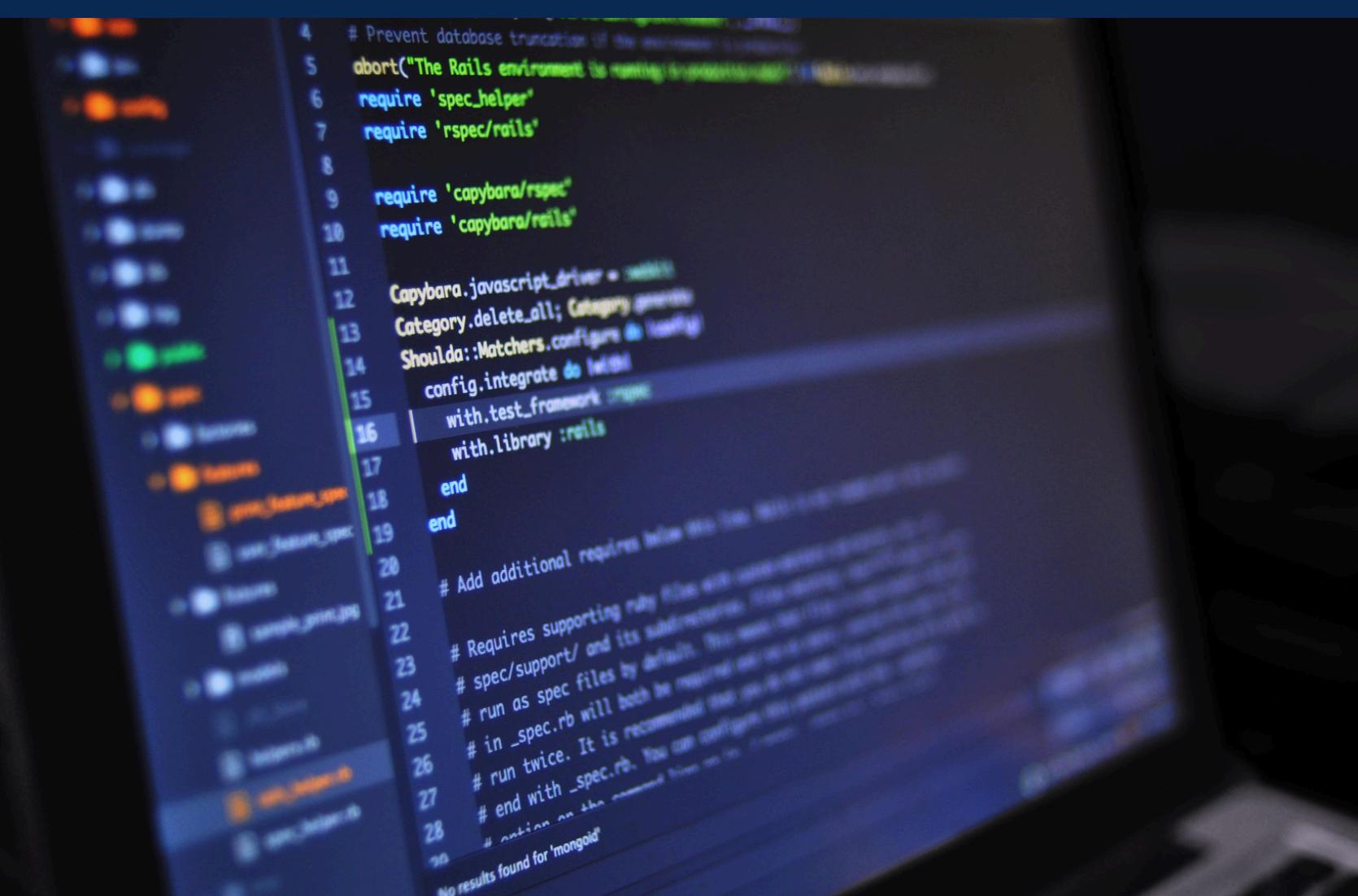
ESTRUCTURA DE DATOS

JUAN SEBASTIÁN OSORIO
FIERRO

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                  |   |   |
                  |   |   |   }}
                </ProductConsumer>
              </div>
            </div>
          </div>
        </React.Fragment>
```

INDICE DE CONTENIDOS

1. COMPARACIÓN DE EDITORES DE TEXTOS E IDE'S	3
2. TIPOS DE LENGUAJE DE PROGRAMACIÓN	5
3. NIVELES DE LENGUAJE DE PROGRAMACIÓN	7
	9
	11



1/ COMPARACIÓN DE EDITORES DE TEXTOS E IDE'S

¿QUE ES UN EDITOR DE TEXTO?

Un editor de texto es un software que permite crear, ver y modificar archivos de texto. A diferencia de los procesadores de texto, como Microsoft Word, que incluyen formato de texto avanzado (negritas, cursivas, fuentes, etc.), un editor de texto generalmente trabaja con texto sin formato (plain text), lo que significa que no tiene formato visual

EJEMPLOS

Algunos ejemplos de editores de texto son:

- Notepad (Bloc de notas) en Windows.
- TextEdit en macOS.
- Vim y Nano en sistemas basados en Unix/Linux.
- Sublime Text, Visual Studio Code y Notepad++, que son más avanzados y ofrecen características adicionales como resaltado de sintaxis, autocompletado, y soporte para múltiples lenguajes de programación.

EJEMPLOS IDE'S

Ejemplos populares de IDEs incluyen:

- Visual Studio para .NET y otros lenguajes.
- Eclipse y IntelliJ IDEA para Java.
- PyCharm para Python.
- Android Studio para el desarrollo de aplicaciones Android

¿QUE ES UN IDE'S

Un IDE (Integrated Development Environment, por sus siglas en inglés) es un entorno de desarrollo integrado que proporciona un conjunto de herramientas completas para facilitar la programación. Un IDE generalmente incluye:

1. Editor de texto: Similar a un editor de texto, pero con características avanzadas como resaltado de sintaxis, autocompletado de código y refactorización.
2. Compilador o intérprete: Herramienta que traduce el código fuente escrito en un lenguaje de programación a código máquina o lo ejecuta directamente.
3. Depurador (Debugger): Permite al programador ejecutar el código paso a paso para encontrar y corregir errores.
4. Integración con sistemas de control de versiones: Como Git, para gestionar cambios en el código y colaborar con otros desarrolladores.
5. Herramientas de construcción: Para automatizar procesos como la compilación, el empaquetado de aplicaciones y la gestión de dependencias.

TIPO DE LENGUAJES DE PROGRAMACION

LENGUAJES NO TIPADOS

- El concepto de "lenguajes no tipados" se refiere a aquellos lenguajes de programación que no utilizan un sistema de tipos formal o en los que el tipo de las variables y expresiones es implícito o inexistente. En realidad, la mayoría de los lenguajes de programación tienen algún tipo de sistema de tipos, pero hay algunos que manejan los tipos de manera muy flexible o casi invisible para el programador.

Importancia y Consideraciones

- Flexibilidad: Los lenguajes que son casi no tipados ofrecen una gran flexibilidad, permitiendo que las variables y operaciones se manejen de manera muy libre.
- Riesgos: Esta flexibilidad también puede llevar a errores que son difíciles de detectar, como manipulaciones de datos incorrectas, debido a la falta de control de tipos.

En resumen, aunque no existen lenguajes "totalmente no tipados" en el sentido moderno, algunos lenguajes manejan los tipos de manera tan implícita o laxa que pueden considerarse "casi no tipados". Esto los hace flexibles pero también potencialmente propensos a errores si no se manejan con cuidado.

LENGUAJES TIPADOS

- Lenguajes tipados estáticos: Java, C, C++, C#, Rust, Haskell.
 - Lenguajes tipados dinámicos: Python, JavaScript, Ruby, PHP, Perl.
 - Lenguajes con tipado fuerte: Java, Python, Rust.
 - Lenguajes con tipado débil: JavaScript, PHP, Perl.
- La elección del tipo de lenguaje depende del contexto de uso y de las preferencias del desarrollador o del equipo de desarrollo.

EJEMPLOS

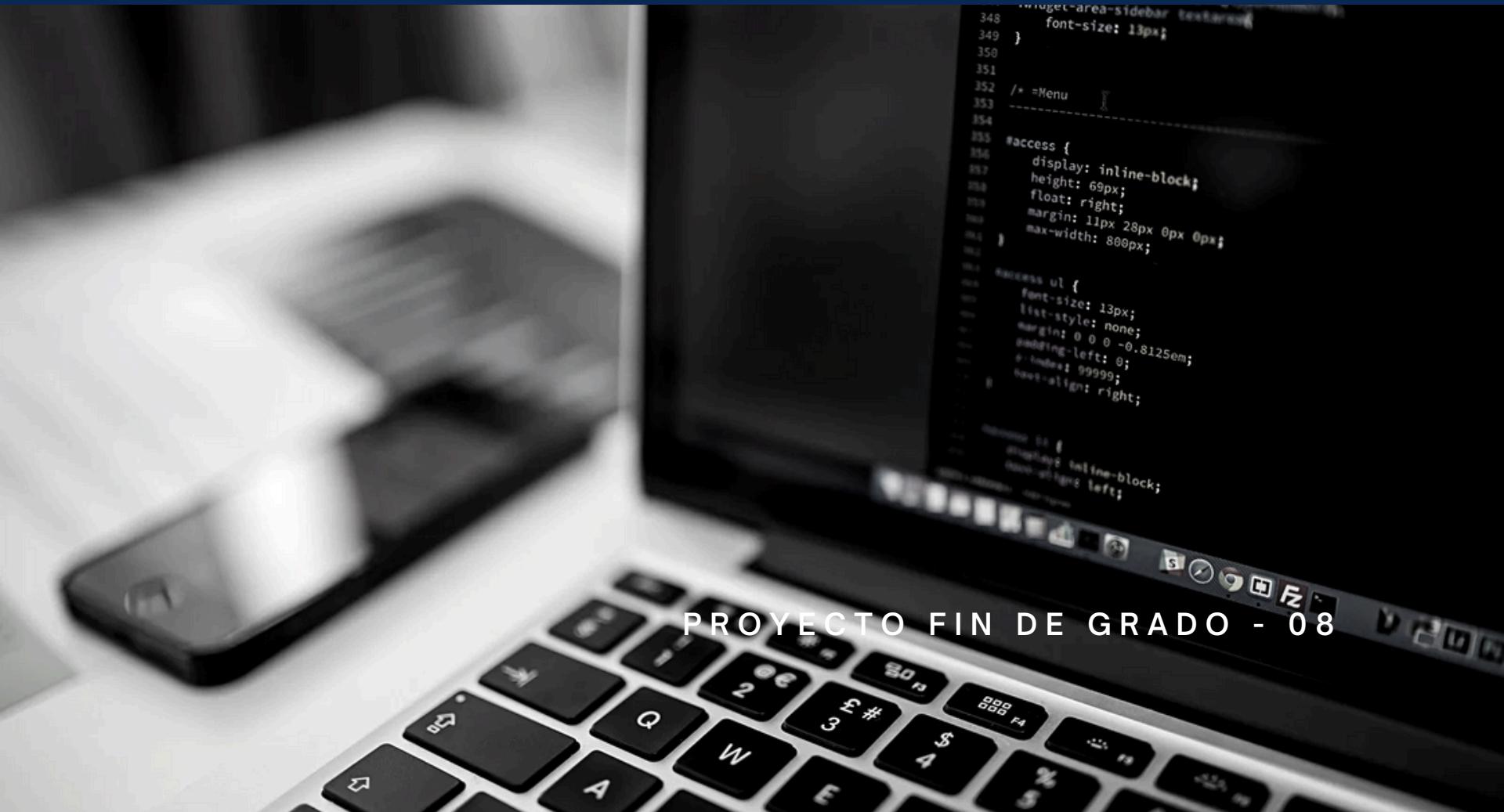
- En los lenguajes con tipado estático, los tipos de las variables se determinan en tiempo de compilación. Esto significa que el tipo de cada variable debe ser conocido antes de que el programa se ejecute.
 - Ejemplos de lenguajes:
 - Java: int numero = 10;
 - C/C++: int numero = 10;
 - C#: int numero = 10;
 - Rust: let numero: i32 = 10;
- }

NIVELES DE LENGUAJE DE PROGRAMACION

NIVELES DE LENGUAJE DE PROGRAMACION

- Bajo Nivel: Lenguaje Máquina, Ensamblador.
- Medio Nivel: C, C++.
- Alto Nivel: Java, Python, JavaScript.
- Muy Alto Nivel (4GL): SQL, MATLAB.
- Nivel Natural (5GL): Prolog, LISP.

Cuanto más alto es el nivel del lenguaje, más fácil es para el humano entender y escribir código, pero a menudo esto significa menos control directo sobre el hardware. Por otro lado, los lenguajes de bajo nivel ofrecen un control más preciso y eficiente, pero requieren un conocimiento más profundo del sistema y son más complejos de programar.



MUCHAS GRACIAS

JUAN SEBASTIAN OSORIO
FIERRO

```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                  |   |   |
                  |   |   |   }}
                </ProductConsumer>
              </div>
            </div>
          </div>
        </React.Fragment>
```