

Projeto de Extensão: Bitcamp (Iniciação à Programação com Linguagem C)

1. Resumo da Proposta

O projeto consiste em um curso presencial de introdução à lógica de programação utilizando a linguagem C, com duração de 25 aulas de 1,5 hora cada. O diferencial da proposta é seu caráter intergeracional, atendendo a um público de idades variadas, e sua execução prática, onde estudantes de Ciência da Computação atuam como instrutores, promovendo a democratização do conhecimento tecnológico e o desenvolvimento de habilidades de ensino.

2. Fundamentação Teórica

A linguagem C, desenvolvida por Dennis Ritchie na década de 1970, permanece como a base fundamental para a compreensão da arquitetura de computadores e sistemas operacionais. Ensinar C no início do aprendizado de programação permite que o aluno compreenda o gerenciamento de memória e a estrutura lógica de baixo nível, facilitando a transição posterior para linguagens de alto nível. Além disso, o ensino de programação estimula o **Pensamento Computacional**, que envolve a decomposição de problemas, reconhecimento de padrões, abstração e design de algoritmos.

3. Justificativa

Vivemos em uma era de alfabetização digital, onde compreender "como as coisas funcionam" por trás das telas é uma competência essencial. Para o público variado (jovens e adultos), o curso oferece inclusão produtiva e estímulo cognitivo. Para a instituição, o projeto reforça o papel do **UNIFUCAMP** na comunidade local. Para os alunos de graduação, a experiência de lecionar consolida o conhecimento técnico através da técnica de Feynman: "quem ensina, aprende duas vezes".

4. Objetivos Gerais

- Proporcionar o primeiro contato da comunidade com o universo da programação.
- Desenvolver o raciocínio lógico e a capacidade de resolução de problemas complexos.

- Capacitar os alunos de Ciência da Computação para a comunicação técnica e a docência.

5. Objetivos Específicos

- Ensinar a sintaxe e a semântica da linguagem C (variáveis, loops, funções).
- Demonstrar a aplicação prática da lógica de programação em pequenos projetos.
- Fomentar a troca de experiências entre diferentes gerações através da tecnologia.
- Entregar um certificado de conclusão para os participantes que atingirem a frequência mínima.

6. Metodologia

O curso será conduzido de forma prática e participativa:

- **Docência Colaborativa:** Os alunos do curso de Ciência da Computação do **UNIFUCAMP** serão divididos em **duplas**. Cada dupla ficará responsável pela preparação e ministração de aulas específicas, sob a supervisão da coordenação.
- **Divisão da Aula:** As sessões de 1h30 serão divididas em 30% de exposição teórica dialogada e 70% de atividades práticas no laboratório de informática.
- **Material de Apoio:** Serão utilizados roteiros de exercícios semanais e desafios de codificação em tempo real.
- **Avaliação:** Será contínua, baseada na participação e na entrega de um mini-projeto final na 25ª aula.

7. Resultados Esperados

- **Para a Comunidade:** Formação de cidadãos capazes de criar pequenos softwares e entender a lógica por trás da tecnologia.

- **Para os Graduandos:** Melhora na oratória, na organização de pensamento e no domínio técnico da linguagem C.
- **Para a Instituição:** Produção de indicadores de extensão e fortalecimento do vínculo entre academia e sociedade.

8. Cronograma

Fase	Atividade
Planejamento	Divisão das duplas de alunos e preparação do material didático.
Execução (Módulo 1-2)	Aulas 01 a 09: Fundamentos e Estruturas Condicionais.
Execução (Módulo 3-4)	Aulas 10 a 19: Laços de Repetição e Vetores/Matrizes.
Execução (Módulo 5)	Aulas 20 a 24: Funções e Structs + Revisão.
Encerramento	Aula 25: Apresentação do Projeto Final e Certificação.

Ementa do Curso: Bitcamp (Iniciação à Programação com Linguagem C)

Objetivo: Introduzir os conceitos de lógica de programação utilizando a linguagem C, capacitando o aluno a desenvolver algoritmos estruturados, manipular dados e resolver problemas lógicos complexos.

Conteúdo Programático

Módulo 1: Primeiros Passos e Estrutura Básica (Aulas 1 a 5)

- **Aula 01:** O que é programação? Compiladores vs. Interpretadores. O "Hello World" e a estrutura de um programa em C.
- **Aula 02:** Variáveis e Tipos de Dados (int, float, char). Como o computador guarda informações.
- **Aula 03:** Entrada e Saída de dados (printf e scanf). Máscaras de formatação (%d, %f, %c).
- **Aula 04:** Operadores Aritméticos e precedência matemática na programação.
- **Aula 05:** Prática de Laboratório: Construção de uma calculadora simples.

Módulo 2: Tomada de Decisão (Aulas 6 a 9)

- **Aula 06:** Expressões Lógicas e Operadores Relacionais (Maior, menor, igual).
- **Aula 07:** Estrutura Condicional Simples e Composta (if e else).
- **Aula 08:** Estruturas Condicionais Encadeadas e Operadores Lógicos (&&, ||, !).
- **Aula 09:** Seleção Múltipla com switch-case.

Módulo 3: Repetição e Automação (Aulas 10 a 14)

- **Aula 10:** Introdução a Laços de Repetição: O conceito de "loop". Estrutura while.
- **Aula 11:** Estrutura de repetição for. Quando usar cada uma.
- **Aula 12:** Estrutura do-while e controle de fluxo (break e continue).
- **Aula 13:** Laços aninhados (um loop dentro do outro) e tabelas.
- **Aula 14:** Prática de Laboratório: Criando um jogo de adivinhação de números.

Módulo 4: Estruturas de Dados Homogêneas (Aulas 15 a 19)

- **Aula 15:** Vetores (Arrays unidimensionais): Armazenando múltiplos valores do mesmo tipo.
- **Aula 16:** Manipulação de Vetores: Busca, soma e média de elementos.
- **Aula 17:** Strings em C: O vetor de caracteres e a biblioteca string.h.
- **Aula 18:** Matrizes (Arrays multidimensionais): Conceito de linhas e colunas.
- **Aula 19:** Prática de Laboratório: Cadastro de notas e nomes de alunos.

Módulo 5: Organização e Modularização (Aulas 20 a 25)

- **Aula 20:** Funções: O que são e por que usar? (Escopo de variáveis).
- **Aula 21:** Funções com retorno e passagem de parâmetros.
- **Aula 22:** Introdução conceitual a Ponteiros (Como a memória funciona).
- **Aula 23:** Structs (Registros): Criando seus próprios tipos de dados.
- **Aula 24:** Revisão Geral: Do algoritmo ao código limpo.
- **Aula 25:** Projeto Final: Desenvolvimento de um sistema de gerenciamento simples (ex: mini-estoque ou agenda).